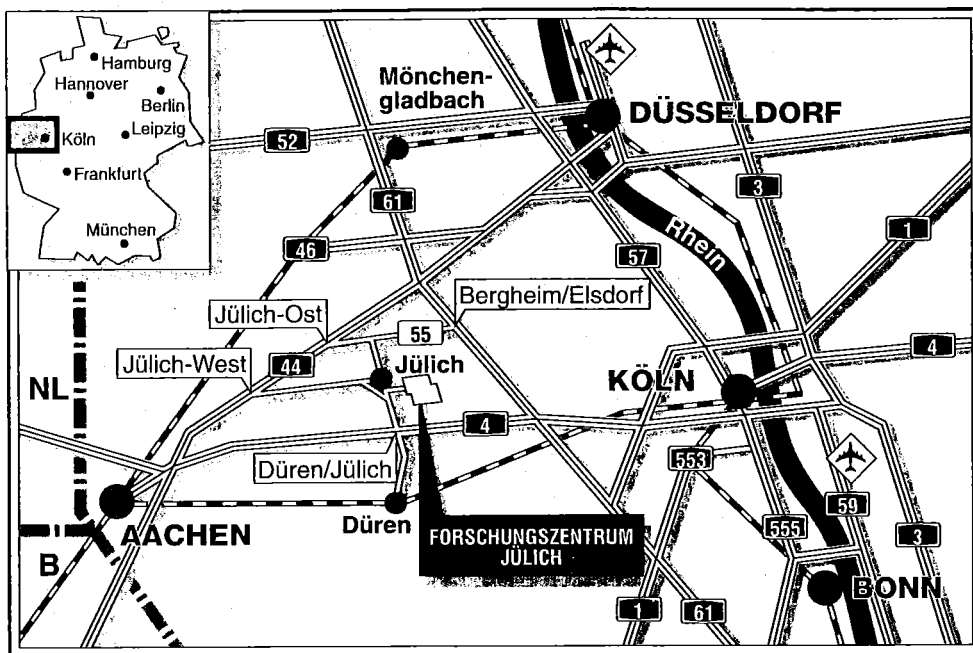




Institut für Sicherheitsforschung und Reaktortechnik

***KONWAR – eine Erweiterung von ATHLET  
zur Berechnung der Kondensation  
in waagerechten Rohren***

*Andreas Schaffrath*



**Berichte des Forschungszentrums Jülich ; 3343**

ISSN 0944-2952

Institut für Sicherheitsforschung und Reaktortechnik Jül-3343

Zu beziehen durch: Forschungszentrum Jülich GmbH · Zentralbibliothek  
D-52425 Jülich · Bundesrepublik Deutschland

☎ 02461/61-6102 · Telefax: 02461/61-6103 · e-mail: [zb-publication@fz-juelich.de](mailto:zb-publication@fz-juelich.de)





# ***KONWAR – eine Erweiterung von ATHLET zur Berechnung der Kondensation in waagerechten Rohren***

*Andreas Schaffrath*

Das diesem Bericht zugrundeliegende Forschungsvorhaben „Berechnung des passiven Notkondensators eines mit Naturumlauf arbeitenden, innovativen Siedewasserreaktors (SWR600) mit ATHLET“ (Förderkennzeichen 15 NU 0948) wurde mit Mitteln des Bundesministeriums für Bildung, Wissenschaft, Forschung und Technologie (BMBF) gefördert. Die Verantwortung für den Inhalt dieser Veröffentlichung liegt beim Autor.



## Kurzfassung

Im Rahmen des Forschungsvorhabens 15NU0948 "Berechnung des passiven Notkondensators eines mit Naturumlauf arbeitenden, innovativen Siedewasserreaktors (SWR600) mit ATHLET" sollen die NOKO-Versuche mit dem Thermohydraulikprogrammsystem ATHLET (Analyse der Thermohydraulik von Lecks und Transienten) nachgerechnet werden. ATHLET wird von der Gesellschaft für Anlagen- und Reaktorsicherheit (GRS) mbH mit der Zielsetzung entwickelt, das gesamte Spektrum von Kühlmittelverlust- und Transientenstörfällen in Leichtwasserreaktoren (LWR) berechnen zu können.

Eine Analyse des in ATHLET enthaltenen Wandkondensationsmodells zeigt, daß dieses nur für die Berechnung der Kondensation von reinen Dämpfen in senkrechten Rohren geeignet ist. Hierbei stellen sich über dem Umfang gleichmäßig dicke Filme ein, bei denen der Schwerkrafteinfluß auf den Film gegenüber dem Einfluß der Phasenreibung vernachlässigt werden kann.

Daher wurde zur Berechnung der Wärmeübergangskoeffizienten bei der Kondensation in horizontalen und leicht gegenüber der Horizontalen geneigten Rohren das Modul KONWAR (Kondensation in waagerechten Rohren) entwickelt und in ATHLET implementiert. KONWAR basiert auf der Strömungskarte von Tandon, mit deren Hilfe die in dem jeweiligen Rohrquerschnitt vorliegende Strömungsform identifiziert wird, und beinhaltet für die einzelnen Strömungsformen verschiedene empirische und halbempirische Korrelationen zur Berechnung der Wärmeübergangskoeffizienten.





## Abstract

Within the framework 15NU0948 "Calculation of the passive emergency condenser of the innovative boiling water reactor (BWR600) operating under natural convection with ATHLET" posttest calculations of NOKO experiments with ATHLET (Analysis of the thermohydraulics of leaks and transients) shall be performed. ATHLET which is being developed by the Gesellschaft für Anlagen- und Reaktorsicherheit (GRS) mbH is intended to cover in a single code the entire spectrum of loss-of-coolant and transient accidents in Light Water Reactors (LWR).

An analysis of the condensation model of ATHLET shows, that the present model is only able to determine heat transfer coefficients for annular flow or laminar films in vertical tubes with equal layer thickness and a non-negligible influence of vapor shear on the condensate film.

For an ATHLET improvement, the modul KONWAR (Kondensation in waagerechten Rohren) has been developed. KONWAR determines heat transfer coefficients during the condensation in horizontal tubes and is based on the flow regime map of Tandon. Therefore KONWAR includes several empirical and semi-empirical correlations for the determination of the heat transfer coefficients.



## Inhaltsverzeichnis

<b>Formelzeichen .....</b>	<b>III</b>
<b>Bild- und Tabellenverzeichnis .....</b>	<b>VI</b>
<b>1 Einleitung .....</b>	<b>1</b>
<b>2 Überblick über die Grundgleichungssysteme von ATHLET .....</b>	<b>3</b>
<b>3 Kondensation und konvektive Wärmeübertragung von Fluiden an feste Strukturen .....</b>	<b>7</b>
3.1 Eignung der Korrelationen in ATHLET zur Berechnung der Wärmeübergangskoeffizienten an der Innenseite der Notkon- densatorrohre .....	9
3.2 Erweiterung des Kondensationsmodells .....	11
3.2.1 Identifizierung der Strömungsform bei der Kondensa- tion in waagerechten Rohren .....	11
3.2.2 Korrelationen zur Bestimmung von Wärmeübergangs- koeffizienten bei der Kondensation in waagerechten Rohren .....	12
3.2.2.1 Sprühströmung .....	14
3.2.2.2 Ringströmung .....	15
3.2.2.3 Schichtenströmung .....	22
3.2.2.4 Übergangsbereich zwischen Ring- und Schich- tenströmung .....	23
3.2.2.5 Blasen-, Pfropfen- und Schwallströmung .....	23
<b>4 Implementierung des erweiterten Kondensationsmodells in ATHLET .....</b>	<b>25</b>
4.1 Funktionen zur Bestimmung von Wärmeübergangskoeffizien- ten .....	28

4.2	Schnittstelle PROP zur Bestimmung der in KONWAR benötigten Wasser-Dampf-Stoffwerte .....	30
4.3	Kopplung von ATHLET und KONWAR.....	31
<b>5</b>	<b>Zusammenfassung und Ausblick.....</b>	<b>33</b>
	<b>Literatur.....</b>	<b>35</b>
	<b>Anhang A: Listing von KONWAR.....</b>	<b>A-1</b>
	<b>Anhang B: Listing der Funktionen zur Berechnung der Wärmeübergangskoeffizienten .....</b>	<b>B-1</b>
	BREBER.....	B-2
	DAMLAM .....	B-4
	DAMTUR .....	B-5
	DAMUEB .....	B-6
	FLUTUR .....	B-8
	FLUUEB .....	B-9
	RING.....	B-10
	RINLAM.....	B-12
	RINTUR.....	B-14
	RINUEB.....	B-16
	SPRUEH.....	B-17
	SCHLAM.....	B-18
	<b>Anhang C: Listing von PROP .....</b>	<b>C-1</b>
	<b>Anhang D: Listing von HCINP .....</b>	<b>D-1</b>
	<b>Anhang E: Unterprogramme MHTCN1 und MHTCN1A .....</b>	<b>E-1</b>

## Formelzeichen

### Lateinische Buchstaben

$A$	Querschnittsfläche	$[m^2]$
$c$	Konstante in Gl. 3.27	$[-]$
$c_p$	spezifische Wärmekapazität	$[J/kgK]$
$D$	Durchmesser	$[m]$
$f$	Korrekturfaktor in Gl. 3.14, 3.18, 3.31	$[-]$
$g$	Gravitationskonstante	$[m/s^2]$
$h$	spez. Enthalpie	$[J/kg]$
$I$	Indikator vgl. Tab. 4.1	$[-]$
$j_D^*$	dimensionslose Dampfgeschwindigkeit nach Wallis gemäß Gl. 3.1	$[-]$
$\dot{m}$	Massenstromdichte	$[kg/m^2s]$
$\dot{M}$	Massenstrom	$[kg/s]$
$n$	Exponent in Gl. 3.27	$[-]$
$N$	Exponent in Gl. 3.25	$[-]$
$p$	Druck	$[Pa]$
$dp/dz$	Druckgradient	$[Pa/m]$
$Q$	Wärmestrom	$[W]$
$r$	spezifische Verdampfungsenthalpie	$[J/kg]$
$T$	Temperatur	$[K]$
$T^+$	dimensionslose Temperatur in Gl. 3.22	$[-]$
$v^*$	Dampfgeschwindigkeit an der Phasengrenze	$[m/s]$
$\dot{x}$	Strömungsdampfgehalt	$[-]$
$x_h$	Enthalpiedampfgehalt	$[-]$
$X$	Zweiphasenparameter in Gl. 3.26	$[-]$

### Griechische Buchstaben

$\alpha$	Wärmeübergangskoeffizient	$[W/m^2K]$
$\beta_A$	Phasenparameter in Gl. 2.1	$[-]$
$\delta^+$	dimensionslose Filmdicke (vgl. Gl. 3.20 bzw. Gl. 3.21)	$[m]$
$\Delta$	Differenz	$[-]$

$\varepsilon$	volumetrischer Dampfgehalt	[-]
$\eta$	dynamische Viskosität	[kg/ms]
$\eta_A$	Wandkondensationsanteil vgl. Gl. 2.1	[-]
$\phi_F^2$	Quotient des Reibungsdruckverlustes der Ein- und Zweiphasenströmung vgl. Gl. 3.24	[-]
$\lambda$	Wärmeleitfähigkeit	[W/mK]
$\nu$	kinematische Viskosität	[m <sup>2</sup> /s]
$\rho$	Dichte	[kg/m <sup>3</sup> ]
$\sigma_O$	Oberflächenspannung	[N/m]
$\zeta$	Widerstandszahl	[-]
$\Psi$	Massenaustauschrate	[kg/s]

## Kennzahlen

$$Re_D = \frac{\dot{x} \dot{m} D}{\eta_D} \quad \text{Reynolds-Zahl des Dampfes}$$

$$Re_{Fi} = \frac{(1-\dot{x}) \dot{m} D}{\eta_{Fi}} \quad \text{Reynolds-Zahl des Kondensatfilms}$$

$$Pr = \frac{\eta c_p}{\lambda} \quad \text{Prandtl-Zahl}$$

## Indizes

### Technische Abkürzungen

D	Dampf
F	Flüssigkeit
Fi	Film
KT	Kondensatfilmtemperatur
lam	laminar
m	Gemisch
S	Sättigung
Sprüh	Sprühströmung
Schicht	Schichtenströmung
Su	Sumpf

tur	turbulent
Well	Wellen
W	Wand
ZP	Zweiphasen

### Korrelationen

Br	Breber
Ko	Kosky und Staub
Nu	Nusselt

### Abkürzungen

#### Technische Begriffe

NOKO	<u>N</u> ot <u>k</u> ondensator
SWR	<u>S</u> ied <u>e</u> wasserreaktor

#### Firmen, Kraftwerke und Institutionen

BMBF	Bundes <u>m</u> inisterium für Bildung, Wissenschaft, <u>E</u> orschung und Technologie
GRS	<u>G</u> esellschaft für Anlagen- und <u>R</u> eaktorsicherheit mbH

#### Computercodes, Module, Programme und Funktionen

ATHLET	Analyse der <u>T</u> hermohydraulik bei <u>L</u> ecks und <u>T</u> ransienten
GCSM	Regel- und Leittechnik- ( <u>G</u> eneral <u>C</u> ontrol and <u>S</u> imulation) <u>M</u> odul
HECU	Wärmetransport- und Wärmeleit- ( <u>H</u> eat <u>c</u> ondu <u>c</u> tion) Modul
KONWAR	<u>K</u> ondensation in <u>w</u> aagerechten <u>R</u> ohren
NEUKIN	<u>N</u> eutronen <u>k</u> inetikmodul
TFD	<u>T</u> hermo <u>f</u> luid <u>d</u> ynamikmodul

## Bildverzeichnis

	Seite
Bild 3.1: Strömungskarte für die Kondensation in horizontalen Rohren nach Tandon [TAT-82] mit der Erweiterung nach Palen [PAJ-79].	13

## Tabellenverzeichnis

	Seite
Tabelle 3.1: Bestimmung des Fluidzustandes in MHTCN1.	7
Tabelle 3.2: Optionen zur Bestimmung von Wärmeübergangskoeffizienten im Unterprogramm MHTCN1 (vgl. [POW-87, LIT-93]).	9
Tabelle 3.3: Werte der Parameter $m$ , $n$ , $c_F$ , $c_D$ , und $N$ in Abhängigkeit von $Re_F$ und $Re_D$ (vgl. [GE-74]).	21
Tabelle 4.1: Optionen zur Bestimmung von Wärmeübergangskoeffizienten in KONWAR.	27



# 1 Einleitung

Der SWR600/1000 ist ein von der Siemens AG entwickeltes, neues innovatives Siedewasserreaktorkonzept. Dieses Konzept ist charakterisiert durch passive Sicherheitssysteme wie z.B. vier Notkondensatoren, vier Gebäudekondensatoren, acht passive Impulsgeber, sechs Flutleitungen, acht Berstmembranen und zwei Abschaltssysteme (vgl. [SIE-93], [SIE-94]).

Zum experimentellen Nachweis der Funktionsweise des Notkondensators wurde am Forschungszentrum Jülich in Kooperation mit der Siemens AG mit Förderung des Bundesministeriums für Bildung, Wissenschaft, Forschung und Technologie (BMBF) sowie den deutschen Energieversorgungsunternehmen der NOKO-Versuchsstand aufgebaut. Der Betriebsdruck der Versuchsanlage beträgt 7,2 MPa, die maximale zur Dampferzeugung dienende Elektrokesselleistung 4 MW. Im NOKO-Versuchsstand wird ein Bündelausschnitt mit acht Rohren, mit dem geplanten Design und Werkstoff des SWR600/1000, eingesetzt. In über 100 Experimenten wurde die Notkondensatorleistung in Abhängigkeit verschiedener Parameter (u.a. Druck und Füllstand im Druckgefäß, Druck, Füllstand und Temperatur im Flutbecken) bestimmt.

Im Rahmen des Forschungsvorhaben 15NU0948 "Berechnung des passiven Notkondensators eines mit Naturumlauf arbeitenden innovativen Siedewasserreaktors" soll das Betriebsverhalten dieses Notkondensators mit dem Thermohydraulikprogrammsystem ATHLET (Analyse der Thermohydraulik von Lecks und Transienten) berechnet werden. ATHLET wird von der Gesellschaft für Anlagen- und Reaktorsicherheit (GRS) mbH mit der Zielsetzung entwickelt, das gesamte Spektrum von Kühlmittelverlust- und Transientenstörfällen in Leichtwasserreaktoren (LWR) berechnen zu können ([GRS-P1], [GRS-P2]).

In dem hier vorliegenden ersten technischen Fachbericht im Rahmen des oben genannten Vorhabens werden zunächst die Grundzüge des ATHLET-Codes einschließlich der hierin enthaltenen thermofluidodynamischen Grundgleichungssysteme vorgestellt (vgl. Kap. 2). Dann folgt die auf einer Analyse des Quellcodes basierende Dokumentation der in ATHLET enthaltenen Modelle zur Beschreibung von Wandkondensation (vgl. Kap. 3) sowie die Überprüfung ihrer Anwendbarkeit bei der Berechnung der Wärmeübertragung im Notkondensator. Für die identifizierten Modellschwächen werden dann Erweiterungs- und Modifi-

kationsvorschläge diskutiert und in Kap. 4 deren Realisierung abschließend detailliert beschrieben. Abschließend folgt in Kap. 5 eine Zusammenfassung und ein Ausblick.

## 2 Überblick über die Grundgleichungssysteme von ATHLET

Die Gesellschaft für Anlagen- und Reaktorsicherheit (GRS) mbH entwickelt das Programmsystem ATHLET (Analyse der Thermohydraulik bei Lecks und Transienten) mit der Zielsetzung, das gesamte Spektrum von Kühlmittelverlust- und Transientenstörfällen in Leichtwasserreaktoren mit einem Code berechnen zu können [TEV-88]. ATHLET besitzt eine streng modulare Struktur. Die Basismodule (Module mit eigenständiger Zeitintegration) dienen zur Simulation

der Thermofluidodynamik (TFD),

des Wärmetransports und der Wärmeleitung (HECU),

der Neutronenkinetik (NEUKIN)

und

der Regel- und Leittechnik (GCSM).

Das Thermofluidodynamikmodul besitzt innerhalb des ATHLET-Codes die zentrale Bedeutung. Die aktuelle ATHLET Version (ATHLET Mod. 1.1) enthält zur Modellierung einer Zweiphasenströmung optional ein 3-, 4- und 5-Gleichungssystem. Die Auswahl des jeweiligen Grundgleichungsmodells nimmt der Anwender in Abhängigkeit von der jeweiligen Problemstellung vor.

Das 3-Gleichungssystem (homogene Massen-, Impuls- und Energiebilanz) modelliert eine Zweiphasenströmung unter Voraussetzung von thermischem und mechanischem Gleichgewicht (d.h. gleiche Geschwindigkeiten in beiden Phasen). Mit dem 4-Gleichungssystem (separierte Massen-, homogene Impuls- und Energiebilanz) kann eingeschränkt thermisches Ungleichgewicht simuliert werden. Hierzu wird eine dominante und eine nicht dominante Phase definiert. Die Annahme, daß die nicht dominante Phase gesättigt ist, erlaubt dann anhand der Energiebilanz, den Zustand der dominanten Phase (unterkühlt, gesättigt oder überhitzt) zu bestimmen. Weiterhin kann mit Hilfe eines mit der Gemischimpulsbilanz gekoppelten Drift-Ansatzes mechanisches Ungleichgewicht modelliert werden [BUM-83]. Erst mit dem Übergang zum 5-Gleichungssystem (separierte Massen- und Energiebilanzen, homogene Impulsbilanz) lassen sich

Strömungen simulieren, bei denen beide Phasen Temperaturen ungleich der Sättigungstemperatur besitzen können. Das mechanische Ungleichgewicht zwischen den Phasen wird wiederum über spezielle Drift-Flux- und Wasserspiegelmodelle erfaßt [STF-89, BAC-91, SCA-93].

Bei thermodynamischem Ungleichgewicht (d.h. bei Verdampfungs- und Kondensationsvorgängen) in einem Kontrollvolumen berechnet ATHLET die Massenaustauschrate mit Hilfe eines Verdampfungsmodells, das bei Kondensationsvorgängen mit negativem Vorzeichen durchlaufen wird. Diese Rate setzt sich aus je einem Anteil für den Phasenübergang im Fluid  $\Psi_F$  und an der Wand  $\Psi_W$  zusammen. Da bei Berechnung des Notkondensators der Zwischenphasenübergang, dessen Modelle ausführlich in [BUM-83] dokumentiert sind, vernachlässigt werden kann, wird nachfolgend nur die Bestimmung der Massenaustauschrate an der Wand  $\Psi_W$ , die sich anhand

$$\Psi_W = \frac{Q \eta_A \beta_A}{r}, \quad (2.1)$$

ergibt, detailliert beschrieben. In obiger Gleichung bezeichnet  $Q$  den zwischen Wand und Fluid ausgetauschten Wärmestrom und  $\eta_A$  den Anteil von  $Q$ , der bei Kondensation direkt an der Wand einen Phasenübergang hervorruft. Bei Verdampfung hingegen besitzt  $\eta_A$  den Wert 1, da die obige Unterteilung nur für Kondensationsvorgänge sinnvoll ist. Der Parameter  $\beta_A$  entspricht bei Kondensationsvorgängen dem Volumendampfgehalt  $\varepsilon$  und bei Verdampfungsvorgängen dem Flüssigkeitsgehalt  $(1-\varepsilon)$ . Weiterhin wird in obiger Gleichung mit  $r$  die Verdampfungsenthalpie bezeichnet.

Der Wärmestrom ergibt sich anhand der Oberfläche der Struktur, dem jeweiligen Wärmeübergangskoeffizienten sowie der Differenz aus der Fluid- und der Wandtemperatur, die beim jeweiligen Aufruf an das Heat-Transfer-Package übergeben werden. In ATHLET sind prinzipiell drei verschiedene Optionen zur Vorgabe bzw. Ermittlung von Wärmeübergangskoeffizienten enthalten. Die erste Option sieht vor, für den Wärmeübergangskoeffizienten einen konstanten Wert innerhalb des Eingabedatensatzes vorzugeben. Bei der zweiten Option können mit Hilfe von GCSM- (General Control and Simulation Modul) Signalen Wärmeübergangskoeffizienten in Abhängigkeit von der Problemzeit spezifiziert werden. Als dritte Option ist in ATHLET die Berechnung von Wärmeüber-

gangskoeffizienten anhand empirischer Korrelationen vorgesehen. Die entsprechenden Korrelationen zur Ermittlung der Wärmeübergangskoeffizienten bei Kondensationsvorgängen sind im Unterprogramm MHTCN1 (vgl. Kap. 3) enthalten.



### 3 Kondensation und konvektive Wärmeübertragung von Fluiden an feste Strukturen

In diesem Kapitel wird zunächst das Kondensationsmodell von ATHLET analysiert. In Kap. 3.1 wird anschließend die Eignung des Modells zur Berechnung der Wärmeübergangskoeffizienten an der Innenseite der Notkondensatorrohre überprüft und Modellschwächen identifiziert. Abschließend wird in Kap. 3.2 eine Vorschlag für eine ATHLET Erweiterung vorgestellt.

In dem Unterprogramm MHTCN1 (vgl. [POW-87, LIT-93]) von ATHLET werden Wärmeübergangskoeffizienten zwischen Wasser, Dampf oder Wasser-Dampf-Gemischen und Strukturen (z.B. ebene Platten, Rohre) für vom Fluid an die Wände übertragene Wärmeströme berechnet. Hierzu wird zunächst der im jeweiligen Kontrollvolumen vorliegende Fluidzustand anhand des enthalpiebezogenen Dampfgehaltes  $x_h$  bzw. des volumetrischen Dampfgehaltes  $\varepsilon$  gemäß Tabelle 3.1 bestimmt. Im einzelnen werden hierbei fünf Bereiche (Flüssigkeit, Zweiphasengemisch, Dampf und 2 Interpolationsbereiche) unterschieden. Die Bereichsgrenzen werden durch Grenzwerte (POL(1) - POL(4)), die in ATHLET mit POL(1) = -0,05, POL(2) = 0, POL(3) = 0,995, POL(4) = 0,999 vorbelegt sind und ggf. durch Vorgabe neuer Werte im Eingabedatensatz überschrieben werden können, festgelegt.

**Tabelle 3.1: Bestimmung des Fluidzustandes in MHTCN1.**

Bereich	Fluidzustand	Bedingung
1	Flüssigkeit	$x_h \leq \text{POL}(1)$
2	Interpolationsbereich zwischen Mode 1 und 5	$\text{POL}(1) < x_h < \text{POL}(2)$
5	Zweiphasengemisch	$x_h \geq \text{POL}(2)$ $\varepsilon \leq \text{POL}(3)$
6	Interpolationsbereich zwischen Mode 5 und 9	$\text{POL}(3) < \varepsilon < \text{POL}(4)$
9	Dampf	$\varepsilon \geq \text{POL}(4)$

Innerhalb der einzelnen Bereiche ergeben sich die Wärmeübergangskoeffizienten als Maximum verschiedener Optionen (Vorgabe von Mindestwerten, empirische Korrelationen). Die Berechnungsoptionen für die Modes 1, 3 und 5 sind in Tabelle 3.2 aufgeführt. Für die Interpolationsgebiete werden die Wärmeübergangskoeffizienten anhand der Korrelationen der benachbarten Modes mit den entsprechenden Grenzwerten für  $x_h$  bzw.  $\varepsilon$  bestimmt und zwischen diesen Werten interpoliert. Die einzelnen in Tabelle 3.2 angegebenen Korrelationen einschließlich ihrer Gültigkeitsbereiche sind ausführlich in [POW-87, LIT-93] beschrieben. Im konvektiven Teil von Bereich 9 (Dampfströmung) steuert der Indikator IHTCI(3) die Auswahl der Korrelation. IHTCI(3) wird von dem aufrufenden Unterprogramm an MHTCN1 übergeben.

### 3.1 Eignung der Korrelationen in ATHLET zur Berechnung der Wärmeübergangskoeffizienten an der Innenseite der Notkondensatorrohre

Bei dem in Kap. 3 beschriebenen Kondensationsmodell werden in den Bereichen 5 und 9 und somit indirekt auch in den Bereichen 2 und 6 die Modelle von Nusselt (laminare Kondensatfilme) [NUW-16] und Carpenter und Colburn (turbulente Kondensatfilme) [CAE-51] zur Bestimmung von Wärmeübergangskoeffizienten bei der Kondensation in Rohren verwendet. Das Modell von **Nusselt** wurde zunächst für die laminare Filmkondensation reiner ruhender Dämpfe an senkrechten ebenen Wänden entwickelt. Hierbei werden aus der Energiebilanz für ein Volumenelement des Kondensatfilms und dem Gleichgewicht zwischen Schwer- und Reibungskräften Korrelationen für die Filmdicke und den Wärmeübergangskoeffizienten abgeleitet. Weitere Vereinfachungen sind, daß sowohl die Dampf- als auch die Wandtemperatur entlang des Rohres konstant ist und der Hauptwärmewiderstand im Kondensatfilm liegt. Der Ansatz von Nusselt läßt sich auch auf verschiedene andere Anwendungsfälle wie z.B. laminare Filmkondensation an bzw. in senkrechten Rohren übertragen, wobei vorausgesetzt wird, daß die Krümmung der Rohrwand vernachlässigbar ist und der Kondensatfilm gleichmäßig über dem Umfang verteilt abfließt.



**Tabelle 3.2: Optionen zur Bestimmung der Wärmeübergangskoeffizienten im Unterprogramm MHTCN1 (vgl. [POW-87, LIT-93]).**

Bereich	Bedingung	Option	Strömungseigenschaft
1		Dittus-Boelter	erzwungene turbulente Flüssigkeitsströmung
		Mc Adams	freie konvektive Flüssigkeitsströmung
		20 W/m <sup>2</sup> K	Mindestwert
5	$T_W > T_S$	Chen	erzwungene turbulente Zweiphasenströmung
		Mc Adams	freie konvektive Zweiphasenströmung
		20 W/m <sup>2</sup> K	Mindestwert
	$T_W \leq T_S$	Nusselt	laminare Filmkondensation
		Carpenter und Colburn	turbulente Filmkondensation
9	$T_W > T_S$ und $IHTCI(3)=1$	Dittus-Boelter	erzwungene turbulente Dampfströmung
	$T_W > T_S$ und $IHTCI(3)=2$	Mc Eligot	erzwungene turbulente Dampfströmung
	$T_W > T_S$	Hausen	erzwungene laminare Dampfströmung
		10 W/m <sup>2</sup> K	Mindestwert
	$T_W \leq T_S$	Nusselt	laminare Filmkondensation
		Carpenter und Colburn	turbulente Filmkondensation

**Carpenter und Colburn** entwickelten ein Modell zur Berechnung von Wärmeübergangskoeffizienten bei der turbulenten Filmkondensation. Hierin unterteilen sie den Kondensatfilm analog zur turbulenten Rohrströmung in drei Bereiche. Dies sind eine wandnahe laminare Unterschicht, eine Übergangsschicht und eine äußere, bis zur Filmoberfläche reichende vollturbulente Schicht. Weiter nehmen Carpenter und Colburn an, daß der gesamte Wärmewiderstand des Kondensatfilms in der dünnen laminaren Unterschicht, deren Dicke experimentell bestimmt wurde, liegt. Filmdicke und Geschwindigkeit in der Unterschicht sind, da hier ein lineares Temperatur- und Geschwindigkeitsprofil vorliegt, in einfacher Weise mit der Wandschubspannung verknüpft [STK-88]. Unter den Annahmen, daß die Schubspannung an der Phasengrenzfläche vereinfacht anhand der für einphasige Strömungen gültigen Korrelation von Blasius bestimmt werden kann und der Dampfgehalt linear mit der Rohrlänge abnimmt, leiteten Carpenter und Colburn nun eine Beziehung für die mittlere Reynolds-Zahl des Kondensatfilms ab und bestimmen hieraus dann eine mittlere Nusseltzahl für das Rohr. Das Modell von Carpenter und Colburn vernachlässigt den Einfluß der Schwerkraft, d.h. im Rohr liegt eine symmetrische turbulente Ringströmung vor. Dies gilt nach [ELN-93] für Dampfgeschwindigkeiten größer 5 m/s. Das Modell von Carpenter und Colburn ist für beliebige Rohranordnungen (d.h. senkrechte, waagerechte oder geneigte Rohre) anwendbar.

ATHLET Mod. 1.1 berechnet bei Kondensation die Wärmeübergangskoeffizienten anhand der beiden oben vorgestellten Modelle und verwendet für die weitere Rechnung den Maximalwert. Das Kondensationsmodell in ATHLET ist in der Lage, die Kondensation von Dämpfen in senkrechten Rohren zu modellieren, da hier der Kondensatfilm als symmetrische laminare oder turbulente Ringströmung abfließt.

Im Gegensatz zur Kondensation in senkrechten Rohren können sich bei der Kondensation in waagerechten Rohren auch andere Strömungsformen als die Ringströmung (z.B. eine Schichten-, Blasen-, Pfropfen-, Schwallströmung) einstellen (vgl. [STK-88]). Der Wärmeübergang hängt maßgeblich von der jeweils vorliegenden Strömungsform ab. Folglich ist zur Bestimmung von Wärmeübergangskoeffizienten entlang der Notkondensatorrohre zunächst die Identifizierung der im jeweiligen Rohrquerschnitt vorliegenden Strömungsform notwendig. Weiterhin sind für alle Strömungsformen, die sich bei der Kondensation in

den waagerechten Rohren einstellen können, Korrelationen zur Bestimmung des Wärmeübergangskoeffizienten in ATHLET zu implementieren.

Die zuvor vorgestellten Modelle zur Bestimmung von Wärmeübergangskoeffizienten bei der Kondensation in waagerechten Rohren von Nusselt sowie von Carpenter und Colburn gelten für die Kondensation von reinen Dämpfen und können folglich nicht den Einfluß nichtkondensierbarer Gase auf die Kondensation berücksichtigen. Da aber die Konzentration der nichtkondensierbaren Gase unter den Einsatzbedingungen des Notkondensators sehr gering ist (d.h. im ppm-Bereich liegt) und diese folglich sowohl die Wärmeübertragung als auch das Strömungsverhalten im Notkondensator nur geringfügig beeinflussen, kann diesbezüglich auf eine spezielle Modellerweiterung verzichtet werden.

## 3.2 Erweiterung des Kondensationsmodells

Eine Zielsetzung dieser Arbeit ist, den ATHLET-Code derart zu erweitern, daß dieser das Betriebsverhalten des Notkondensators - speziell die Kondensation von Dämpfen in waagerechten Rohren - berechnen kann. Gemäß den obigen Ausführungen ist hierzu die Ermittlung der im jeweiligen Querschnitt der waagerechten Rohre vorliegenden Strömungsform (vgl. Kap. 3.2.1) und die Bereitstellung von Korrelationen zur Bestimmung des Wärmeübergangskoeffizienten für die einzelnen Strömungsformen (vgl. Kap. 3.2.2) notwendig. Im Anschluß an die Beschreibung der Modellerweiterung wird dann detailliert auf die Realisierung der Erweiterungen in ATHLET eingegangen.

### 3.2.1 Identifizierung der Strömungsform bei der Kondensation in waagerechten Rohren

Tandon et. al. geben in [TAT-82] eine Strömungskarte zur Bestimmung der Strömungsform bei der Kondensation in waagerechten Rohren an. Hierin wird die von Wallis [WAG-69] eingeführte dimensionslose Dampfgeschwindigkeit  $j_D^*$

$$j_D^* = \frac{\dot{x}\dot{M}}{A[gD\rho_D(\rho_F - \rho_D)]^{0,5}} \quad (3.1)$$

über dem aus dem volumetrischen Flüssigkeitsgehalt und dem volumetrischen Dampfgehalt in einem Querschnitt gebildeten Quotienten  $(1-\epsilon)/\epsilon$  aufgetragen (vgl. Bild 3.2 sowie Bild 1.1). In obiger Gleichung bezeichnet  $\dot{x}$  den Strömungsdampfgehalt, der als Quotient des Dampfmassenstromes  $\dot{M}_D$  und des Gesamtmassestromes  $\dot{M}$

$$\dot{x} = \frac{\dot{M}_D}{\dot{M}} \quad (3.2)$$

definiert ist,  $A$  die Querschnittsfläche des Rohres mit dem Durchmesser  $D$ ,  $g$  die Erdbeschleunigung,  $\rho_D$  die Dampf- und  $\rho_F$  die Kondensatdichte. Weiterhin kennzeichnet  $\epsilon$  den volumetrischen Dampfgehalt in einem Kontrollvolumen.

Aus [TAT-82] lassen sich folgende Grenzen für die einzelnen Strömungsformen entnehmen:

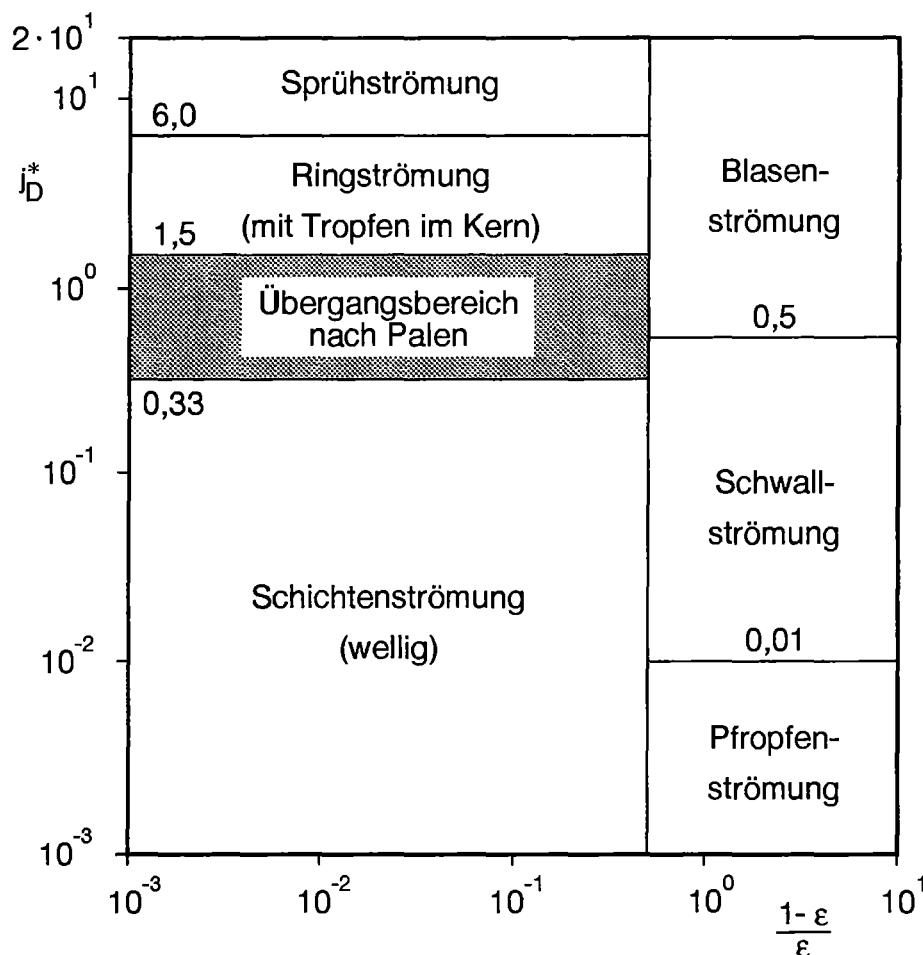
Schichtenströmung	$j_D^* < 1$	und	$(1-\epsilon)/\epsilon \leq 0,5$ ,
Ringströmung	$1 \leq j_D^* \leq 6$	und	$(1-\epsilon)/\epsilon \leq 0,5$ ,
Sprühströmung	$j_D^* > 6$	und	$(1-\epsilon)/\epsilon \leq 0,5$ ,
Blasenströmung	$j_D^* > 0,5$	und	$(1-\epsilon)/\epsilon > 0,5$ ,
Schwallströmung	$0,01 \leq j_D^* \leq 0,5$	und	$(1-\epsilon)/\epsilon > 0,5$ ,
Pfropfenströmung	$j_D^* < 0,01$	und	$(1-\epsilon)/\epsilon > 0,5$ .

Darüber hinaus gibt Palen [PAJ-79] an, daß zwischen Schichten- und Ringströmung ein Übergangsbereich existiert. Die Grenzen für diesen Übergangsbereich liegen bei  $0,33 \leq j_D^* \leq 1,5$  und  $(1-\epsilon)/\epsilon \leq 0,5$  (vgl. Bild 3.1).

### 3.2.2 Korrelationen zur Bestimmung von Wärmeübergangskoeffizienten bei der Kondensation in waagerechten Rohren

Anhand der Strömungskarte von Tandon ergeben sich bei der Kondensation in waagerechten Rohren 6 verschiedene Strömungsformen. Weiterhin befindet sich zwischen der Ring- und Schichtenströmung noch ein ausgeprägter Übergangsbereich, in dem die Wärmeübergangskoeffizienten mittels Interpolation ermittelt werden. Nachfolgend werden in diesem Kapitel für die einzelnen Strömungsformen (Sprüh-, Ring-, Schichten-, Pfropfen-, Schwall- und Blasen-

strömung), die für die Erweiterung des Kondensationsmodells in ATHLET ausgewählten Korrelationen vorgestellt. Die Auswahl der Korrelationen erfolgt u.a. anhand ihres Gültigkeitsbereiches und berücksichtigt die in vergleichenden Literaturstudien [LUK-80, SHM-81, REU-841] und wärmetechnischen Handbüchern [GE-76, GE-90, VDI-88] angegebenen Empfehlungen. Mit Ausnahme der Ringströmung wurde angestrebt, für jede Strömungsform eine Korrelation zur Berechnung von Wärmeübergangskoeffizienten zu verwenden. Bei der Ringströmung ist dies nicht möglich, da hierbei der Kondensatfilm sowohl laminar als auch turbulent strömen kann und daher unterschiedliche Typen von Modellen zur Berechnung von Wärmeübergangskoeffizienten zu verwenden sind.



**Bild 3.1: Strömungskarte für die Kondensation in horizontalen Rohren nach Tandon [TAT-82] mit der Erweiterung nach Palen [PAJ-79].**

### 3.2.2.1 Sprühströmung

Die Erweiterung des Kondensationsmodells enthält aus Vollständigkeitsgründen ebenfalls ein Modell zur Berechnung von Wärmeübergangskoeffizienten bei Sprühströmung, obwohl sich diese Strömungsform bei dem geplanten Design nicht innerhalb der Notkondensatorrohre einstellt. Das nachfolgend vorgestellte Modell wurde von Soliman [SOH-86] entwickelt und an experimentellen Daten für Wasserdampf und Kältemittel (R-11, R-13) überprüft. Hierbei geht Solimann von der aus Beobachtungen abgeleiteten Annahme aus, daß beide Phasen als homogene Mischung mit thermodynamischen Gleichgewicht nebeneinander strömen. Der Wärmeübergangskoeffizient bei Sprühströmung  $\alpha_{\text{Sprüh}}$  berechnet sich über

$$\alpha_{\text{Sprüh}} = 0,00345 \text{Re}_m^{0,9} \text{Pr}_D^{0,33} \left( \frac{r}{c_{p,D} (T_S - T_W)} \right)^{-0,33} \quad (3.3)$$

In der obigen Definition bezeichnet  $\text{Re}_m$  die Reynolds-Zahl des Gemisches, die sich über

$$\text{Re}_m = \frac{\dot{m} D}{\eta_m} \quad (3.4)$$

mit  $\dot{m}$  als Massenstromdichte,  $D$  als Rohrdurchmesser und  $\eta_m$  als dynamischer Viskosität des Gemisches ergibt. Die dynamische Viskosität des Gemisches  $\eta_m$  berechnet sich über

$$\frac{1}{\eta_m} = \frac{\dot{x}}{\eta_D} + \frac{(1-\dot{x})}{\eta_F}, \quad (3.5)$$

mit  $\dot{x}$  als Strömungsdampfgehalt,  $\eta_D$  als dynamischer Viskosität des Dampfes und  $\eta_F$  als dynamischer Viskosität des Films. Weiterhin bezeichnet  $\text{Pr}_D$  in der Korrelation von Solimann die Prandtl-Zahl des Dampfes,  $r$  die Verdampfungsenthalpie,  $c_{p,D}$  die spezifische Wärmekapazität des Dampfes,  $T_W$  die Wandtemperatur und  $T_S$  die Sättigungstemperatur.

### 3.2.2.2 Ringströmung

Bei der Kondensation in waagerechten Rohren zählt die Ringströmung neben der Schichtenströmung zu den am häufigsten vorkommenden Strömungsformen. Für Ringströmungen liegen umfangreiche experimentelle Daten vor, die u.a. auch zur Ableitung von empirischen oder halbempirischen Korrelationen zur Bestimmung von Wärmeübergangskoeffizienten herangezogen werden. Bei der Ringströmung ist zwischen laminaren, laminar welligen und turbulenten Kondensatfilmen zu unterscheiden.

Die Berechnung von Wärmeübergangskoeffizienten bei Ringströmungen mit laminaren bzw. laminar welligen Kondensatfilmen basiert auf dem Modell von Nusselt [NUW-16], das zur Berücksichtigung von Oberflächenwellen, Dampfüberhitzung, Kondensatunterkühlung und temperaturabhängigen Stoffwerten im Kondensatfilm geeignet modifiziert wird. Bei der Berechnung von Wärmeübergangskoeffizienten bei Ringströmungen mit turbulenten Filmen wird u.a. zwischen rein empirischen Korrelationen (z.B. Akers [AKW-60]) und aus der Grenzschichttheorie für turbulente Rohrströmungen abgeleiteten Modellen (z.B. Traviss [TRD-78], Carpenter und Colburn [CAE-51], Kosky und Staub [KOP-71], Shah [SHM-79, SHM-81]) unterschieden. Die meisten dieser Korrelationen sind hinsichtlich ihrer Wiedergabegenauigkeit und ihres Parameterbereiches sehr beschränkt. Für die ATHLET-Erweiterung wird daher das Modell von Kosky und Staub ausgewählt.

Bevor nachfolgend die Modelle zur Berechnung von Wärmeübergangskoeffizienten bei laminaren bzw. laminar welligen und turbulenten Kondensatfilmen vorgestellt werden, folgt zunächst die Diskussion der Kriterien zur Ermittlung des jeweiligen Filmzustandes (laminar, laminar wellig, turbulent). Die Einteilung der einzelnen Bereiche wird anhand der Reynolds-Zahl des Kondensatfilms  $Re_{Fi}$  vorgenommen, die definiert ist als:

$$Re_{Fi} = \frac{(1 - \dot{x}) \dot{m} D}{\eta_{Fi}} \quad (3.6)$$

In obiger Gleichung bezeichnet  $\dot{x}$  den Strömungsdampfgehalt,  $\dot{m}$  die Massenströmdichte,  $D$  den Rohrdurchmesser und  $\eta_{Fi}$  die dynamische Viskosität des Films. Der Übergang vom laminaren zum laminar welligen Film erfolgt, falls die

Reynolds-Zahl des Kondensatfilms den Grenzwert  $Re_{Well}$  und der Übergang von laminar welligen zu turbulenten Kondensatfilmen den Grenzwert  $Re_{krit}$  überschreitet.

Im Bereich sehr kleiner Film-Reynolds-Zahlen (Kutateladse:  $Re_{Fi} < 1$  [KUS-63], Elsner:  $Re_{Fi} < 8$  [ELN-93] oder Renz:  $Re_{Fi} < 10$  [REU-84]) ist der Kondensatfilm wellenfrei. Grimley [GRS-45] beobachtete, daß bei  $Re_{Fi} \geq Re_{Well}$  Wellen oder kleine Rippen auf der Filmoberfläche erschienen. Diese verbessern den Wärmeübergang um bis zu 40%. Nach Grimley ergibt sich  $Re_{Well}$  anhand:

$$Re_{Well} = 0,392 \left[ \left( \frac{\sigma_O}{\rho_{Fi} g} \right)^{0,5} \left( \frac{g}{\nu_{Fi}^2} \right)^{0,33} \right]^{0,75} . \quad (3.7)$$

In der obigen Gleichung, die von van der Walt und Kröger mit Hilfe einer Störungsrechnung bestätigt wurde [WAJ-74], bezeichnet  $\sigma_O$  die Oberflächenspannung,  $g$  die Erdbeschleunigung,  $\rho_{Fi}$  die Dichte und  $\nu_{Fi}$  die kinematische Viskosität des Kondensatfilms. Der Einfluß von Oberflächenwellen auf den Wärmeübergangskoeffizienten ist nur bei laminaren Filmen von Bedeutung, da bei turbulenten Filmen der Einfluß der turbulenten Impuls- und Energieaustauschmechanismen auf den Wärmeübergangskoeffizienten gegenüber dem Einfluß der Oberflächenwellen dominiert.

Die Film-Reynolds-Zahl  $Re_{Fi}$  des Umschlagpunktes zwischen laminar welligen und turbulenten Kondensatfilmen wird nach [VDI-88] in Abhängigkeit von der Prandtl-Zahl des Kondensatfilms  $Pr_{Fi}$  angegeben:

$$Re_{krit} = \frac{100}{Pr_{Fi}} . \quad (3.8)$$

Die Prandtl-Zahl des Kondensatfilms  $Pr_{Fi}$  ist definiert als

$$Pr_{Fi} = \left( \frac{\eta c_p}{\lambda} \right)_{Fi} , \quad (3.9)$$

mit  $\lambda$  als Wärmeleitfähigkeit,  $c_p$  als spezifische Wärmekapazität und  $\eta$  als dynamischer Viskosität des Films. Allerdings ist diese Korrelation nicht unumstritten.



ten, da die Prandtl-Zahl nicht in die Berechnung der Nusselt-Zahl bei laminaren Kondensatfilmen einfließt.

Für **laminar wellenfreie Filme**, d.h. für  $Re_{Fi} < Re_{Well}$ , zeigte Schmidt [SCT-51], daß der Wärmeübergangskoeffizient bei der Kondensation in waagerechten Rohren anhand des Modells von Nusselt berechnet werden kann. Der Wärmeübergangskoeffizient bei laminarer Filmkondensation  $\alpha_{Nu}$  ergibt sich somit zu:

$$\alpha_{Nu} = 1,1025 \lambda_{Fi} Re_{Fi}^{-0,33} \left( \frac{\rho_{Fi}(\rho_{Fi} - \rho_D)g}{\eta_{Fi}^2} \right)^{0,33} . \quad (3.10)$$

Die Stoffwerte sind in obiger Gleichung bei der effektiven Kondensatfilmtemperatur  $T_{KT}$  einzusetzen

$$T_{TK} = T_W + f_{KT} (T_D - T_S) . \quad (3.11)$$

In obiger Gleichung wird mit  $T_W$  die Wand-, mit  $T_D$  die Dampf- und mit  $T_S$  die Sättigungstemperatur bezeichnet. Weiterhin berücksichtigt der Faktor  $f_{KT}$  die Krümmung des Temperaturprofils im Kondensatfilm. Dieser besitzt nach Minkowycz und Sparrow den Wert 0,31 [COL-82].

Um die **Unterkühlung des Kondensats bzw. eine Überhitzung des Dampfes** bei der Kondensation reiner Dämpfe in waagerechten oder leicht geneigten Rohren berücksichtigen zu können, erweiterte Rohsenow [ROW-56] den Ansatz von Nusselt. Hierzu ersetzt er in dem Modell von Nusselt die Verdampfungsenthalpie durch die Enthalpiedifferenz  $\Delta h^*$

$$\Delta h^* = c_{p,D} (T_D - T_S) + r + 0,68 c_{p,F} (T_S - T_F) , \quad (3.12)$$

mit  $T$  als Temperatur und  $c_p$  als spezifische Wärmekapazität der einzelnen Phasen. Der Index D kennzeichnet hierbei die Dampfphase und der Index F die Flüssigkeitsphase. Bei technischen Anwendungen sind in der Regel allerdings die Dampfüberhitzung und die Kondensatunterkühlung vernachlässigbar.

In der Nusselt'schen Wasserhauttheorie wird vorausgesetzt, daß die Stoffwerte des Kondensatfilms unabhängig von der Filmtemperatur sind. Diese Annahme gilt jedoch nur bei schwach gekrümmten Temperaturprofilen über dem Kon-

densatfilm. Bei Differenzen zwischen der Wand- und der Sättigungstemperatur des Dampfes von kleiner als 50 K ist der Fehler bei der Berechnung von Wärmeübergangskoeffizienten unter Vernachlässigung der **Temperaturabhängigkeit der Stoffwerte** kleiner als 3% [BAH-94]. Bei großen Differenzen zwischen der Wand- und der Sättigungstemperatur des Dampfes kann die Temperaturabhängigkeit der dynamischen Viskosität und der Wärmeleitfähigkeit durch Multiplikation des nach Nusselt berechneten Wärmeübergangskoeffizienten mit dem Korrekturfaktor  $f_T$

$$\alpha = \alpha_{Nu} f_T \quad (3.13)$$

berücksichtigt werden. Der Korrekturterm  $f_T$  ergibt sich nach Stephan [STK-88] anhand:

$$f_T = \frac{1 + \eta^*}{10(1 + \lambda^*)^3} \left[ 5 + \lambda^* (14 + 11 \lambda^*) + \frac{\lambda^*}{\eta^*} (1 + 4 \lambda^* + 5 \lambda^{*2}) \right]. \quad (3.14)$$

In obiger Gleichung bezeichnet  $\eta^*$  das Verhältnis der dynamischen Viskositäten des Kondensatfilms bei der Sättigungstemperatur (Index S) bzw. der Wandtemperatur (Index W)

$$\eta^* = \frac{\eta_S}{\eta_W} \quad (3.15)$$

und analog hierzu  $\lambda^*$  das entsprechende Verhältnis der Wärmeleitfähigkeiten:

$$\lambda^* = \frac{\lambda_S}{\lambda_W}. \quad (3.16)$$

Der Temperatureinfluß auf die Dichte des Kondensatfilms ist nach [STK-88] vernachlässigbar.

Das Modell von Nusselt kann ebenfalls für die Berechnung von Wärmeübergangskoeffizienten bei **laminar welligen Filmen**, d.h.  $Re_{Well} < Re_{Fi} < Re_{krit}$  verwendet werden. Hierzu wird der anhand des Nusselt-Ansatzes ermittelte Wärmeübergangskoeffizient  $\alpha_{Nu}$  mit einem Korrekturfaktor zur Berücksichtigung der Wellenbildung  $f_{Well}$  multipliziert. Es gilt:

$$\alpha_{Well} = \alpha_{Nu} f_{Well}. \quad (3.17)$$

In der Literatur sind verschiedene empirische Korrelationen zu finden, die den Einfluß der Wellen auf den Wärmeübergangskoeffizienten beschreiben. Nach Zazuli [KUS-63] kann für  $Re_{Fi} > Re_{Well}$  der Korrekturfaktor zur Bestimmung des Welleneinflusses  $f_{Well}$  in Abhängigkeit von der Reynolds-Zahl anhand

$$f_{Well} = 0,8 Re_{Fi}^{0,11} \quad (3.18)$$

berechnet werden.  $Re_{Well}$  ergibt sich hierbei anhand des oben beschriebenen Ansatzes von Grimley (vgl. Gl. 3.7).

Für **turbulente Ringströmungen**, d.h.  $Re_{Fi} > Re_{krit}$  wird der Wärmeübergangskoeffizient anhand des Modells vom Kosky und Staub [KOP-71] bestimmt. In diesem Modell werden die Wärmeübergangskoeffizienten in Abhängigkeit von der dimensionslosen Temperatur  $T^+$ , die sich wiederum als Funktion der dimensionslosen Schichtdicke des Kondensatfilms  $\delta^+$  ergibt, und der Dampfgeschwindigkeit an der Phasengrenze  $v^*$  bestimmt, mit:

$$\alpha_{Ko} = \frac{\rho_{Fi} v^* c_{p,Fi}}{T^+ (\delta^+)}. \quad (3.19)$$

In der obigen Gleichung bezeichnen  $\rho_{Fi}$  die Dichte des Kondensats und  $c_{p,Fi}$  dessen spezifische Wärmeleitfähigkeit. Die dimensionslose Kondensatfilmschichtdicke  $\delta^+$  ergibt sich in Abhängigkeit von der Reynolds-Zahl des Kondensatfilms. Für  $Re_{Fi} < 1000$  wird  $\delta^+$  über

$$\delta^+ = 0,0504 (0,5 Re_{Fi})^{0,5} \quad (3.20)$$

und für  $Re_{Fi} \geq 1000$  über

$$\delta^+ = 0,0504 (Re_{Fi})^{7/8} \quad (3.21)$$

bestimmt. Bei der Berechnung von  $T^+$  ist je nach Größe von  $\delta^+$  eine Fallunterscheidung vorzunehmen. Es gilt:

$$T^+(\delta^+) = \delta^+ Pr_{Fi}, \quad \text{für } \delta^+ \leq 5,$$

$$T^+(\delta^+) = 5 \left[ \text{Pr}_{\text{Fi}} + \ln \left( 1 + \text{Pr}_{\text{Fi}} (0,2 \delta^+ - 1) \right) \right], \quad \text{für } 5 \leq \delta^+ \leq 30$$

und

$$T^+(\delta^+) = 5 \left[ \text{Pr}_{\text{Fi}} + \ln (1 + 5 \text{Pr}_{\text{Fi}}) + 0,495 \ln \left( \frac{1}{30} \delta^+ \right) \right], \quad \text{für } \delta^+ \geq 30. \quad (3.22)$$

Die Dampfgeschwindigkeit an der Phasengrenze  $v^*$  wird von Kosky und Staub über

$$v^* = \sqrt{\frac{D}{4 \rho_F} \left( \frac{dp}{dz} \right)_{\text{Reib,ZP}}} \quad (3.23)$$

mit  $D$  als Rohrdurchmesser,  $\rho_F$  als Fluidichte und  $(dp/dz)_{\text{Reib,ZP}}$ , dem Reibungsdruckverlust der Zweiphasenströmung, ermittelt.

Der Reibungsdruckverlust ergibt sich gemäß Wallis [WAG-69] über das heterogene Modell, d.h. beide Phasen strömen getrennt voneinander mit unterschiedlicher Geschwindigkeit, zu:

$$\phi_F^2 = \frac{(dp/dz)_{\text{Reib,ZP}}}{(dp/dz)_{\text{Reib,F}}}. \quad (3.24)$$

In obiger Gleichung bezeichnet  $\phi_F^2$  den aus Reibungsdruckverlust der Zweiphasenströmung  $(dp/dz)_{\text{Reib,ZP}}$  und der einphasigen Flüssigkeitsströmung  $(dp/dz)_{\text{Reib,F}}$  gebildeten Quotienten. Dieser ergibt sich in Abhängigkeit vom Lockhart-Martinelli-Parameter  $X$  [LOR-49], [MAR-69] zu

$$\phi_F^2 = \left[ 1 + \left( \frac{1}{X} \right)^{2/N} \right]^N, \quad (3.25)$$

mit

$$X^2 = \frac{\zeta_F}{\zeta_D} \frac{\rho_D}{\rho_F} \left( \frac{1-x}{x} \right)^2. \quad (3.26)$$

Nach Einsetzen des Blasius'schen Gesetzes für die Widerstandsbeiwerte  $\zeta_F$  und  $\zeta_D$

$$\zeta_K = \frac{c_K}{Re_K^{\eta_K}} \quad (3.27)$$

mit  $c$  als Konstante (vgl. Tab. 3.3) und  $Re$  als Reynolds-Zahl der Phase  $K$  (der Phasenindex  $K = D$  kennzeichnet die Dampfphase, der Phasenindex  $K = F$  die Flüssigkeitsphase) folgt:

$$X^2 = \left[ \frac{Re_D^{\eta_D}}{Re_F^{\eta_F}} \right] \frac{c_F \rho_D}{c_D \rho_F} \left( \frac{1-X}{X} \right)^2. \quad (3.28)$$

Die Reynolds-Zahl des Dampfes  $Re_D$  wird hierin über

$$Re_D = \frac{\dot{x} \dot{m} D}{\eta_D} \quad (3.29)$$

bestimmt. In den obigen Gleichungen bezeichnet  $D$  den Rohrdurchmesser,  $\dot{m}$  die gesamte Massenstromdichte,  $\dot{x}$  den Dampfgehalt der Strömung,  $\eta$  die dynamischen Viskositäten und  $\rho$  die Dichten der einzelnen Phasen. Hierbei kennzeichnet der Index  $F$  das Kondensat und der Index  $D$  den Dampf. Alle weiteren Parameter, d.h.  $\eta_D$ ,  $\eta_F$ ,  $c_F$ ,  $c_D$ , und  $N$  sind in Abhängigkeit von  $Re_D$  und  $Re_F$  Tabelle 3.3 zu entnehmen. Der Umschlagpunkt von laminarer zur turbulenter Strömung erfolgt bei  $Re_F$  bzw.  $Re_D$  von 2000. Für Drücke kleiner 1,1 MPa besitzt  $N$  bei Kondensation den Wert 5,13 [GE-74].

**Tabelle 3.3: Werte der Parameter  $m$ ,  $n$ ,  $c_F$ ,  $c_D$ , und  $N$  in Abhängigkeit von  $Re_F$  und  $Re_D$  (vgl. [GE-74]).**

Parameter	$Re_F > 2000$ $Re_D > 2000$	$Re_F < 2000$ $Re_D > 2000$	$Re_F > 2000$ $Re_D < 2000$	$Re_F < 2000$ $Re_D < 2000$
$\eta_D$	0,25	0,25	1,0	1,0
$\eta_F$	0,25	1,0	0,25	1,0
$c_F$	0,079	16,0	0,079	16,0
$c_D$	0,079	0,079	16,0	16,0
$N$	4,0 ( $p \geq 1,1$ MPa) 5,13 ( $p < 1,1$ MPa)	3,5	3,5	2,75

### 3.2.2.3 Schichtenströmung

Bei einer Schichtenströmung setzt sich der in einem Rohrquerschnitt übertragene Wärmestrom aus zwei Anteilen zusammen. Dies sind der durch den Kondensatfilm am Umfang und der durch die Kondensatschicht am Boden (Sumpf) übertragene Wärmestrom. Der Wärmeübergangskoeffizient der Kondensatschicht am Boden ist in guter Näherung gegenüber dem des Films vernachlässigbar.

Die meisten der in der Literatur angegebenen Ansätze zur Bestimmung von Wärmeübergangskoeffizienten bei Schichtenströmung basieren auf dem Nusselt-Modell [NUW-16], wobei der nach Gl. 3.10 bestimmte Wärmeübergangskoeffizient  $\alpha_{Nu}$  um einen Korrekturfaktor zur Berücksichtigung des Sumpfeinflusses  $f_{Su}$  erweitert wird. Es gilt:

$$\alpha_{Schicht} = \alpha_{Nu} f_{Su} \quad (3.30)$$

Viele Autoren z.B. Nilsson [SHM-81], Breber [BRG-80], Chato [CHJ-62], Kern [KED-50], Kroger [KRD-76] betrachten nur einen über der gesamten Rohrlänge gemittelten Wärmeübergangskoeffizienten. Folglich ist  $f_{Su}$  eine Konstante. Baehr [BAH-94], Jaster und Kosky [JAH-76] und Rufer und Kezios [RUC-66] bestimmen zur Berücksichtigung des Sumpfeinflusses auf den Wärmeübergangskoeffizienten lokale bzw. über einzelne Abschnitte der Kondensatorrohre gemittelte Korrekturfaktoren für den anhand des Modells von Nusselt [NUW-16] bestimmten Wärmeübergangskoeffizienten. In allen Fällen ergibt sich der Korrekturfaktor anhand des lokalen bzw. über einem Rohrabschnitt gemittelten Wert des volumetrischen Dampfgehaltes  $\varepsilon$ . Nach Jaster und Kosky bzw. Rufer und Kezios gilt:

$$f_{Su} = \varepsilon^{0,75}. \quad (3.31)$$

Die Verbesserung der Wärmeübertragung durch das Auftreten von Oberflächenwellen ist analog zu Kap. 3.2.2.2 zu berücksichtigen.

### 3.2.2.4 Übergangsbereich zwischen Ring- und Schichtenströmung

Innerhalb des Übergangsbereiches zwischen Ring- und Schichtenströmung schlägt Palen [PAJ-79] vor, den Wärmeübergang durch Überlagerung der Wärmeübergangskoeffizienten für Schichten und Ringströmung gemäß:

$$\alpha_{\text{Üb}} = \alpha_{\text{Schicht}} + \frac{(j_D^* - 1,5)}{1,17} (\alpha_{\text{Ring}} - \alpha_{\text{Schicht}}) . \quad (3.32)$$

In obiger Gleichung kennzeichnet  $\alpha_{\text{Schicht}}$  den Wärmeübergangskoeffizient bei Schichtenströmung, der sich anhand der in Kap. 3.2.2.3 angegebenen Gleichungen ergibt und  $\alpha_{\text{Ring}}$  den Wärmeübergangskoeffizient bei Ringströmung. Die entsprechenden Gleichungen zur Bestimmung von  $\alpha_{\text{Ring}}$  sind in Kap. 3.2.2.2 angegeben.

### 3.2.2.5 Blasen-, Pfropfen- und Schwallströmung

Für den Bereich der Blasen-, Pfropfen- und Schwallströmung sind zwar verschiedene Modelle z.B. Breber [BRG-79] veröffentlicht worden, jedoch steht bislang ein Vergleich mit experimentellen Daten aus [VDI-88]. Das Modell von Breber basiert auf den Korrelationen zur Bestimmung des Wärmeübergangskoeffizienten bei einphasiger laminarer oder turbulenter Flüssigkeitsströmung. Ähnlich wie beim Modell von Kosky und Staub (vgl. Kap. 3.2.2.2) berücksichtigt Breber den Phasenübergang durch Multiplikation des Wärmeübergangskoeffizienten der Einphasenströmung mit der 0,45-ten Potenz des aus dem Reibungsdruckverlustes der Zweiphasenströmung und der einphasigen Flüssigkeitsströmung gebildeten Quotienten. Dieser ist analog zu Kap. 3.2.2.2 zu bestimmen. Es gilt:

$$\alpha_{\text{Br}} = \alpha_{\text{F}} (\varphi_{\text{F}}^2)^{0,45} \quad (3.33)$$

Mit  $\alpha_{\text{F}}$  wird in obiger Gleichung der Wärmeübergangskoeffizient der einphasigen Strömung bezeichnet, der nach Breber sowohl im laminaren als auch im turbulenten Fall anhand der Korrelationen von Sieder-Tate zu bestimmen ist.

Die Korrelation von Sieder-Tate für laminare Flüssigkeitsströmungen [SIE-36] lautet

$$\alpha_{\text{lam}} = 1,86 \frac{\lambda_F}{D} \left( \text{Re}_F \text{Pr}_F \frac{D}{l} \right)^{0,33} \left( \frac{\eta_F}{\eta_W} \right)^{0,14} \quad (3.34)$$

und eignet sich zur Berechnung von Wärmeübergangskoeffizienten bei Flüssigkeiten mit Prandtl-Zahlen  $\text{Pr}_F \approx 1$ . Weiterhin bezeichnet in obiger Gleichung  $l$  die Rohrlänge,  $\eta_F$  die dynamische Viskosität der Flüssigkeit und  $\eta_W$  die dynamische Viskosität der Flüssigkeit bei der Temperatur der Rohrwand. Alle übrigen Stoffwerte sind bei der Flüssigkeitstemperatur zu berechnen. Die Prandtl-Zahl  $\text{Pr}_F$  der Flüssigkeit ist definiert als

$$\text{Pr}_F = \left( \frac{\eta c_p}{\lambda} \right)_F \quad (3.35)$$

mit  $\lambda$  als Wärmeleitfähigkeit,  $c_p$  als spezifischer Wärmekapazität und  $\eta$  als dynamischer Viskosität der Flüssigkeit. Analog zu Tab. 1.3 wird nun der Umschlagpunkt von laminarer zu turbulenter Strömung bei  $\text{Re}_F = 2000$  angenommen.

Die Korrelation von Sieder-Tate für turbulente Flüssigkeitsströmungen [SIE-36] basiert auf der Dittus-Boelter Korrelation [DIF-30], die um einen Korrekturfaktor zur Berücksichtigung der Zähigkeitsunterschiede aufgrund des Temperaturprofils in der Flüssigkeit ergänzt wird. Die Korrelation von Sieder-Tate für turbulente Flüssigkeitsströmungen lautet:

$$\alpha_{\text{tur}} = 0,024 \frac{\lambda_F}{D} \text{Re}_F^{0,8} \text{Pr}_F^{0,33} \left( \frac{\eta_F}{\eta_W} \right)^{0,14} \quad (3.36)$$

Hierbei sind die Stoffwerte wiederum bis auf  $\eta_W$  bei der Flüssigkeitstemperatur einzusetzen. Bezugstemperatur für die dynamische Viskosität  $\eta_W$  ist weiterhin die Temperatur der Rohrwand.



## 4 Implementierung des erweiterten Kondensationsmodells in ATHLET

Die in Kapitel 3.2 beschriebene Erweiterung des Kondensationsmodells ist in dem Modul KONWAR (Kondensation in waagerechten Bohren) realisiert. Nachfolgend wird nun der Aufbau und die Struktur dieses Moduls, die Funktionen von KONWAR (vgl. Kap. 4.1) einschließlich der Schnittstelle zur Bestimmung der in KONWAR benötigten Wasser-Dampf-Stoffwerte (vgl. Kap. 4.2) und die Kopplung mit ATHLET (vgl. Kap. 4.3) detailliert beschrieben.

Das Modul KONWAR besteht zum einen aus dem Steuerprogramm KONWAR, (das Listing von KONWAR befindet sich in Anhang A) in dem die Strömungskarte von Tandon einschließlich des Übergangsbereiches zwischen Ring- und Schichtenströmung (vgl. Kap. 3.2.1) programmiert ist. Weiterhin sind in Modul KONWAR zahlreiche Unterprogramme für die Berechnung der Wärmeübergangskoeffizienten für die einzelnen Strömungsformen (vgl. Tab. 4.1) und die Ermittlung der in KONWAR benötigten Wasser-Dampf-Stoffwerte sowie eine Interpolationsroutine enthalten.

Das Steuerprogramm TANDON besitzt die drei Funktionen Datenaustausch mit ATHLET, Bereitstellung sämtlicher in KONWAR benötigten Wasser-Dampf-Stoffwerte und Verzweigung zu den Unterprogrammen, in denen die Wärmeübergangskoeffizienten berechnet werden. Das Unterprogramm KONWAR ist die Schnittstelle zwischen dem Modul KONWAR und dem Unterprogramm MHTCN1 in ATHLET und beinhaltet daher die Übergabe sämtlicher zur Berechnung der Wärmeübergangskoeffizienten bei der Kondensation in waagerechten Rohren benötigten Parameter. Unmittelbar nach der Variablendeklaration erfolgt in KONWAR der Aufruf der Schnittstelle PROP, über die zentral sämtliche in KONWAR benötigten Wasser-Dampf-Stoffwerte bestimmt werden (vgl. Kap. 4.2).

Im Anschluß an die Stoffwertbestimmung werden zunächst sämtliche zur Fallunterscheidung benötigten Kennzahlen (z.B. Reynolds-Zahl des Dampfes  $Re_D$ , der Flüssigkeit  $Re_F$  sowie des Films  $Re_{Fi}$ , die dimensionslose Dampfgeschwindigkeit nach Wallis und das Verhältnis von Flüssigkeits- zu Dampfgehalt in

einem Kontrollvolumen) bestimmt. Anhand dieser Parameter erfolgen dann die Identifizierung der aktuell vorliegenden Strömungsform (vgl. Kap. 3.2.1) und die Verzweigung in die entsprechenden Bereiche von KONWAR. Hier sind die Modelle zur Berechnung der Wärmeübergangskoeffizienten (vgl. Tab. 4.1) enthalten. Die Kennzeichnung der Strömungsformen erfolgt durch Indikatoren. Diese werden zu den mit dem Faktor 100 multiplizierten (nachfolgend in Tab. 4.1 mit I bezeichneten) Indikatoren der Bereiche von MHTCN1, aus denen KONWAR aufgerufen wird, hinzuaddiert und während einer Rechnung für jedes Kontrollvolumen sowie zu jedem Ausgabezeitpunkt ins Standardausgabefile ausgegeben.

Die Übergänge zwischen den einzelnen Strömungsformen sind aus programmtechnischen Gründen nicht als starre Grenzen sondern in Form von Interpolationsbereichen modelliert. Sämtliche Interpolationen werden mit Hilfe der aus ATHLET übernommenen linearen Interpolationsroutine SPOLY [GRS-P2] durchgeführt. Weiterhin wurden ebenfalls Übergangsbereiche zwischen der Ein- und Zweiphasenströmung definiert. Dies gilt sowohl für den Anfang der Notkondensatorrohre, wo der Dampf zu kondensieren beginnt, als auch für die Strömungen nach der vollständigen Kondensation des Dampfes, wenn eine reine einphasige Flüssigkeitsströmung vorliegt. Der einphasige Wärmeübergang wird mit den aus ATHLET übernommenen Modellen beschrieben:

<b>laminare Dampfströmung:</b>	Hausen [HAH-43],
<b>turbulente Dampfströmung:</b>	Dittus-Boelter [DIF-30],
<b>laminare Flüssigkeitsströmung:</b>	Sieder-Tate [SIE-36],
<b>turbulente Flüssigkeitsströmung:</b>	Dittus-Boelter [DIF-30], Sieder-Tate [SIE-36].

Zwischen den laminaren und turbulenten Strömungsbereichen ist jeweils ein Übergangsbereich definiert. Die Identifizierung des jeweiligen Strömungszustandes geschieht mit Hilfe der Reynolds-Zahl (vgl. Tab. 4.1). Die oben aufgeführten Modelle für den einphasigen Wärmeübergang werden an dieser Stelle nicht extra beschrieben, da sie bereits in den Unterprogrammen MTHCN1 ent-

halten und somit sowohl bekannt als auch bereits ausführlich dokumentiert sind [vgl. LIK-75, POW-87, LIT-93]. Nachfolgend werden nun die einzelnen Unterprogramme von KONWAR, in denen die Wärmeübergangskoeffizienten bestimmt werden, die Schnittstelle PROP und die hiervon aufgerufenen Stoffwertefunktionen vorgestellt.

**Tabelle 4.1: Optionen zur Bestimmung von Wärmeübergangskoeffizienten in KONWAR.**

Bereich	Bedingung	Option	Strömungseigenschaft
I+1	$Re_F > 2000$ $\varepsilon = 0$	Max. (Dittus-Boelter, Sieder Tate)	erzwungene turbulente Flüssigkeitsströmung
	$Re_F \leq 2000$ $\varepsilon = 0$	Sieder-Tate	erzwungene laminare Flüssigkeitsströmung
I+10	$j_n^* \geq 6$ und $(1-\varepsilon)/\varepsilon \leq 0,5$	Solimann	Sprühströmung
I+30	$0,33 \leq j_n^* \leq 6$ $(1-\varepsilon)/\varepsilon \leq 0,5$ $Re_{Fi} < Re_{krit}$	Nusselt	laminare Ringströmung
	$1,5 \leq j_n^* \leq 6$ $(1-\varepsilon)/\varepsilon \leq 0,5$ $Re_{Fi} \geq Re_{krit}$	Kosky und Staub	turbulente Ringströmung
I+40	$0,33 \leq j_n^* \leq 1,5$ $(1-\varepsilon)/\varepsilon \leq 0,5$	Palen	Übergangsbereich zwischen Ring- und Schichtenströmung
I+50	$j_n^* \leq 0,33$ $(1-\varepsilon)/\varepsilon \leq 0,5$	Rufer und Kezios	Schichtenströmung
I+60 I+70 I+80	$(1-\varepsilon)/\varepsilon \geq 0,5$	Breber	Blasenströmung, Pfropfenströmung, Schwallströmung
I+9	$Re_D \leq 2000$ $\varepsilon = 1$	Hausen	laminare Dampfströmung
	$Re_D > 2000$ $\varepsilon = 1$	Dittus-Boelter	turbulente Dampfströmung

## 4.1 Funktionen zur Bestimmung von Wärmeübergangskoeffizienten

Das Modul KONWAR enthält insgesamt 13 Funktionen, in denen die Berechnung der Wärmeübergangskoeffizienten für die einzelnen Strömungsformen bei der Kondensation in waagerechten Rohren und für einphasige konvektive Wärmeübertragung enthalten ist. Diese werden nachfolgend kurz vorgestellt. Ein Listing sämtlicher Funktionen befindet sich in Anhang A.

### **BREBER**

In der Funktion BREBER werden die Wärmeübergangskoeffizienten bei Pfropfen-, Schwall- und Blasenströmung anhand des in Kap. 3.2.2.2 beschriebenen Modells von Breber berechnet. In BREBER ist weiterhin die Ermittlung der im Modell benötigten einphasigen Wärmeübergangskoeffizienten nach Sieder-Tate, und des Reibungsdruckverlustes der Zweiphasenströmung anhand des heterogenen Modells von Lockhart und Martinelli enthalten.

### **DAMLAM**

In der Funktion DAMLAM werden die Wärmeübergangskoeffizienten bei laminarer Dampfströmung anhand der Korrelation von Hausen berechnet.

### **DAMTUR**

In der Funktion DAMTUR werden die Wärmeübergangskoeffizienten bei turbulenter Dampfströmung anhand der Korrelation von Dittus-Boelter berechnet.

### **DAMUEB**

In der Funktion DAMUEB werden die Wärmeübergangskoeffizienten im Übergangsbereich von laminaren zu turbulenten Dampfströmungen berechnet. Hierzu bestimmt DAMUEB durch Aufruf von DAMLAM und DAMTUR die Wärmeübergangskoeffizienten am Rande des Übergangsbereiches und ermittelt in Abhängigkeit von der jeweiligen Reynolds-Zahl mittels Interpolation den Wärmeübergangskoeffizienten im Übergangsbereich.

**FLULAM**

In der Funktion FLULAM werden die Wärmeübergangskoeffizienten bei laminaren Flüssigkeitsströmungen anhand der Korrelationen von Sieder-Tate berechnet.

**FLUTUR**

In der Funktion FLUTUR werden die Wärmeübergangskoeffizienten bei turbulenten Flüssigkeitsströmungen anhand der Korrelationen von Dittus-Boelter sowie Sieder-Tate berechnet. Für die weitere Rechnung wird dann das Maximum aus beiden Optionen an das FLUTUR aufrufende Unterprogramm zurückübergeben.

**FLUUEB**

In der Funktion FLUUEB werden die Wärmeübergangskoeffizienten im Übergangsbereich von laminaren zur turbulenten Flüssigkeitsströmungen berechnet. Hierzu bestimmt FLUUEB durch Aufruf von FLULAM und FLUTUR die Wärmeübergangskoeffizienten am Rande des Übergangsbereiches und bestimmt in Abhängigkeit von der jeweiligen Reynolds-Zahl mittels Interpolation den Wärmeübergangskoeffizienten im Übergangsbereich.

**RING**

In der Funktion RING wird die Berechnung der Wärmeübergangskoeffizienten bei Ringströmung gesteuert. In RING sind die Kriterien zur Unterscheidung von laminaren, laminar welligen und turbulenten Kondensatfilmen enthalten. Zwischen den Strömungsformen sind Übergangsbereiche definiert. Danach werden zur Berechnung der Wärmeübergangskoeffizienten für jeden dieser drei Fälle separate Funktionen (RINLAM, RINTUR, RINUEB) aufgerufen.

**RINLAM**

In der Funktion RINLAM werden die Wärmeübergangskoeffizienten bei laminaren Ringströmungen anhand der in Kap. 3.2.2.2 beschriebenen Modifikation des Modells von Nusselt berechnet. RINLAM kann den Einfluß von Dampfüberhitzung, Kondensatunterkühlung, temperaturabhängiger Stoffwerte im Kondensatfilm und Wellenbildung auf der Filmoberfläche auf die Wärmeübertragung berücksichtigen.

**RINTUR**

In der Funktion RINTUR werden die Wärmeübergangskoeffizienten bei der Ringströmung mit turbulenten Kondensatfilmen anhand des in Kap. 3.2.2.2 beschriebenen Modells von Kosky und Staub berechnet. Ferner ist in RINTUR das heterogene Modell zur Ermittlung der Reibungsdruckverluste der Zweiphasenströmung nach Lockhart und Martinelli enthalten.

**RINUEB**

In der Funktion RINUEB werden die Wärmeübergangskoeffizienten im Übergangsbereich von laminar welligen zu turbulenten Ringströmungen bestimmt.

**SPRUEH**

In der Funktion SPRUEH werden die Wärmeübergangskoeffizienten bei Sprühströmung anhand des in Kap. 3.2.2.1 beschriebenen Modells von Soliman berechnet.

**SCHLAM**

In dem Unterprogramm SCHLAM werden die Wärmeübergangskoeffizienten bei Schichtenströmung anhand des modifizierten Modells von Nusselt (vgl. Kap. 3.2.3.3) beschrieben. SCHLAM berücksichtigt den Einfluß von Wellenbildung auf der Kondensatfilmoberfläche auf die Wärmeübertragung.

## **4.2 Schnittstelle PROP zur Bestimmung der in KONWAR benötigten Wasser-Dampf-Stoffwerte**

Innerhalb von KONWAR werden an zahlreichen Stellen Wasser-Dampf-Stoffwerte benötigt. Dies sind u.a. spezifische Volumina, kinematische und dynamische Viskositäten, Wärmeleitfähigkeiten und Enthalpien der einzelnen Phasen bei verschiedenen Referenztemperaturen (z.B. bei Sättigungstemperatur, mittlerer Filmtemperatur). Die Stoffwerteberechnung erfolgt zentral für das Modul KONWAR unmittelbar nach dessen Aufruf über die ebenfalls neu geschaffene Schnittstelle PROP (vgl. Anhang C). Die Stoffwerte werden mit Hilfe der bereits in ATHLET enthaltenen Stoffwertefunktionen MPATS, MPTAF, MPHFG berechnet. Die in den jeweiligen Funktionen berechneten Parameter sind nachfolgender Übersicht zu entnehmen:

**MPATS:** Sättigungstemperatur,

**MPTAF:** spez. Volumen, Enthalpie, Wärmekapazität, Wärmeleitfähigkeit, dynamische Viskosität, Oberflächenspannung,

**MPAHFG:** Verdampfungsenthalpie.

Zum anderen werden in PROP aber auch weitere, aus den oben aufgeführten Parametern abgeleitete Stoffwerte (z.B. Dichten, Prandtl-Zahlen, kinematische Viskositäten) bestimmt. Für sämtliche an bzw. von PROP übergebene Daten können umfangreiche Kontrollausgaben aktiviert werden.

### 4.3 Kopplung von ATHLET und KONWAR

Das Modul KONWAR wird - sofern dies der Anwender wünscht - vom Unterprogramm MHTCN1A bei Kondensation (d.h. Bereich 5 oder 9) anstelle des standardmäßig in ATHLET enthaltenen Wandkondensationsmodells aufgerufen. Hierzu müssen die HECU-Objekte, für die KONWAR angewendet werden soll mit der Buchstabenkombination "WR" beginnen. Beim Einlesen der HECU-Objekte in dem Unterprogramm HCINP (vgl. Anhang D) werden daraufhin die Objektnamen überprüft und in dem COMMON-Block *ANDREA* ein entsprechender Indikator (1 Verwendung von KONWAR, 0 Verwendung des Original-Wandkondensationsmodells) für die einzelnen Objekte belegt. Durch Laden von *ANDREA* in MHTCN1A (vgl. Anhang E) wird dann der weitere Programmablauf gesteuert. Das Unterprogramm MHTCN1A stellt eine überarbeitete und erweiterte Version des Unterprogramms MHTCN1 dar und wird in diesem aufgerufen, wenn der Indikator für das jeweilige HECU-Objekt im COMMON-Block *ANDREA* den Wert 1 besitzt.





## 5 Zusammenfassung und Ausblick

Zur Nachrechnung der NOKO-Experimente wurde der ATHLET-Code so erweitert, daß dieser in der Lage ist, die Kondensation in waagerechten und leicht gegenüber der Horizontalen geneigten Rohren zu simulieren. Hierzu wurde ATHLET mit dem Modul KONWAR gekoppelt. KONWAR basiert auf der Strömungskarte von Tandon für die Kondensation in waagerechten Rohren und beinhaltet ferner die Erweiterung von Palen (Definition des Übergangsbereiches zwischen Ring- und Schichtenströmung). Für jede Strömungsform beinhaltet KONWAR empirische oder halbempirische Korrelationen zur Berechnung des Wärmeübergangskoeffizienten. Im einzelnen sind dies das Model von Soliman für die Sprühströmung, das Model von Kosky und Staub für die turbulente Ringströmung, das Model von Nusselt für die laminare Ringströmung, das Model von Jaster und Kosky bzw. Rufer und Kezios für die Schichtenströmung sowie das Model von Breber für die Blasen-, Pfropfen- und Schwallströmung. Für laminare Ringströmungen kann darüber hinaus der Einfluß von Oberflächenwellen, Dampfüberhitzung, Kondensatunterkühlung sowie temperaturabhängiger Stoffwerte über Korrekturfaktoren berücksichtigt werden.

Nachfolgend ist geplant, die um KONWAR erweiterte Version des ATHLET Codes anhand der NOKO-Experimente zu validieren. Darüber hinaus wäre es sinnvoll, KONWAR anhand weiterer Experimente z.B. die HORUS-Experimente in Zittau [LIW-95] zu validieren und den erweiterten ATHLET zur Nachrechnung der Kondensation in den Heizrohren liegender Dampferzeuger von WWER-Anlagen einzusetzen.



## Literatur

- [AKW-60] W. W. Akers, H. F. Rosson. *"Condensation inside a horizontal tube"*, Chem. Eng. Prog. Symp. Ser. 56 (1960), Nr. 30, S. 145 - 149.
- [BAC-91] C. Bals, G. Lerchl, D. Loy. *"Entwicklung einer ATHLET-Version mit zwei Energiegleichungen"*. GRS-A-1830, August 1991.
- [BAH-94] H. D. Baehr. *"Wärme- und Stoffübertragung"*, Springer-Verlag Berlin Heidelberg New York London Paris Tokyo Hong Kong Barcelona Budapest 1994.
- [BRG-79] G. Breber, J. W. Palen, J. Taborek. *"Prediction of horizontal tube-wise condensation of pure components using flow regime criteria"*, Advances in Enhanced Heat Transfer ASME, 18th National Heat Transfer Conference, 6.-8.8.1979 San Diego California, S. 1 - 8.
- [BRG-80] G. Breber, J. W. Palen, J. Taborek. *"Prediction of horizontal tube-side condensation of pure components using flow regime criteria"*, ASME Journal Heat Transfer 102 (1980), S. 471 - 476.
- [BUM-83] M. J. Burwell, D. Enix, E. Hofer, G. Lerchl, W. Pointner, F. Steinhoff, I. Vojtek, K. Wolfert. *"DRUFAN-01/MOD Program Description - Vol. III: Model Description"*. GRS-A-846, Juli 1983.
- [CAE-51] E. F. Carpenter, A. P. Colburn. *"The effect of vapor velocity on condensation inside tubes"*, Proc. of the General Discussion of Heat Transfer, Inst. Mech. Eng., New York 1951, S. 20 - 26.
- [COL-82] J. G. Collier. *"Convective boiling and condensation"*, Mc Graw-Hill International Book Company, New York 1982.
- [CHJ-62] J. C. Chato. *"Laminar condensation inside horizontal tubes"*, ASHRAE J. Vol. 4 (1962), Nr. 2, S. 52 - 60.
- [DIF-30] F. W. Dittus, L. M. K. Boelter. *"Heat transfer in automobile radiators of the tubular type"*, Uni. of Cal. Publications in Engineering, Vol. 2 (1930), Nr. 13, S. 443 - 461.

- [ELN-93] N. Elsner, S. Fischer, J. Huhn. *"Grundlagen der technischen Thermodynamik"*, Akademie Verlag, Berlin 1993.
- [GE-74] General Electric. *"Heat Transfer Data Book: Fluid Flow Division"*, Section 402.4, Genium Publishing 1974.
- [GE-76] General Electric. *"Heat Transfer Data Book: Heat Transfer Division"*, Section 506.3, Genium Publishing 1976.
- [GE-90] General Electric. *"Heat Transfer Data Book: Heat Transfer Division"*, Section 506.3, Genium Publishing 1990.
- [GRS-45] S. S. Grimley. *"Liquid flow conditions in packed towers"* Trans. Inst. Chem. Eng. (London) 23 (1945), S. 228 - 235.
- [GRS-P1] G. Lerchl, H. Austregesilo. *"ATHLET Mod. 1.1 Cycle C - User's Manual"*, Gesellschaft für Anlagen- und Reaktorsicherheit (GRS) mbH. GRS - P-1/Vol. 1, Oktober 1995.
- [GRS-P2] H. Austregesilo, H. Deitenbeck. *"ATHLET Mod. 1.1 Cycle C - Programmer's Manual"*, Gesellschaft für Anlagen- und Reaktorsicherheit (GRS) mbH. GRS - P-1/Vol. 2, Oktober 1995.
- [HAH-43] H. Hausen, *"Darstellung des Wärmeüberganges in Rohren durch verallgemeinerte Potenzbeziehungen"*, VDI-Zeitschrift, 1943.
- [HAW-96] W. Hartel, J. Mattern. *"Der SWR1000 - Technik, Sicherheit, Wirtschaftlichkeit"*, Jahrestagung Kerntechnik, Fachsitzung "Reaktorsicherheit, 21.-23.5.1996 in Mannheim.
- [JAH-76] H. Jaster, P. G. Kosky. *"Condensation heat transfer in a mixed flow regime"*, Int. J. Heat Mass Transfer 19 (1976), S. 95 - 99.
- [KED-50] D. Q. Kern. *"Process heat transfer"*, Mc Graw-Hill Book Company, New York Toronto London 1950.
- [KOP-71] P. G. Kosky, W. F. Staub. *"Local condensing heat transfer coefficients in the annular flow regime"*, AIChE Journal 17 (1971), S. 1037 - 1043.

- [KRD-76] D. C. Kroger. *"Laminar condensation heat transfer inside inclined tubes"*, Chem. Eng. Symp. Ser. 73 (1976), S. 256 - 260.
- [KUS-63] S. S. Kutateladse. *"Fundamentals of heat transfer"*, Edward Arnold, London 1963.
- [LIK-75] K. J. Liesch, G. Raemhild, K. Hofmann. *"Zur Berechnung des Wärmeüberganges und der kritischen Heizflächenbelastung im Hinblick auf besondere Verhältnisse in den Kühlkanälen eines DWR bei schweren Störfällen"*, Lehrstuhl für Reaktordynamik und Reaktorsicherheit, Technische Universität München, Juni 1975.
- [LIT-93] T. Linnemann, A. Schaffrath, U. Brockmeier. *"Analyse und Dokumentation des Unterprogrammblocks zur Ermittlung von Wärmeübergangskoeffizienten im Programmsystem ATHLET"*, RUB E-42, Mai 1993.
- [LIW-95] W. Lischke, S. Alt, B. Kraus, A. Fjodorow. *"Experimentelle und analytische Untersuchungen des thermohydraulischen Verhaltens der Heizrohre liegender Dampferzeuger von WWER-Anlagen unter Störfalbedingungen"*, Jahrestagung Kerntechnik, Nürnberg, 16.-18. Mai 1995.
- [LOR-49] R. W. Lockart, R. C. Martinelli. *"Proposed correlation of data for isothermal two-phase, two-component flow in pipes"*, Chemical Engineering Progress Vol. 45 (1949), Nr. 1, S. 39.
- [LUK-80] K. Lucas. *"Kondensation in Rohren - eine Analyse praktischer Rechenmethoden"*, Chemie-Ing.-Techn. 52 (1980), Nr. 7.
- [MAR-69] R. C. Martinelli, D. B. Nelson, N. Y. Schenectady. *"Prediction of pressure drop during forced circulation boiling of water"*, Trans. Am. Soc. Mech. Eng. Vol. 70 (1948), S. 695-702.
- [NUW-16] W. Nusselt. *"Oberflächenkondensation des Wasserdampfes"*, Zeitschrift des Vereins Deutscher Ingenieure 27 (1916), Band 60, S. 541 - 546 und S. 569 - 575.

- [PAJ-79] J. W. Palen, G. Breber, J. Taborek. *"Prediction of flow regimes in horizontal tube condensation"*, Heat Transfer Eng. 1 (1979), S. 47 - 57.
- [POW-87] W. Pointner. *"Interim Program Description Part 3: Wall Heat Transfer Package in DRUFAN-02 and its Application to LOBI and LOFT Experiments"* GRS-A-1325, Februar 1987.
- [REU-84] U. Renz. *"Maßnahmen zur Verbesserung des Wärmeübergangs bei der Kondensation"*, Chem.-Ing. Technik 56 (1984), Nr. 11, S. 807 - 818.
- [ROW-56] W. M. Rohsenow. *"Heat transfer and temperature distribution in laminar film condensation"*, Trans. Am. Soc. Mech. Eng. Journal Heat Transfer 78 (1956), S. 1645 - 1648.
- [RUC-66] C. E. Rufer, S. P. Kezios. *"Analysis of two-phase, one-component stratified flow with condensation"*, Trans. Am. Soc. Mech. Eng. Journal Heat Transfer 88 (1966), S. 265.
- [SCA-93] A. Schaffrath, H. Sprünken, A.-K. Krüsenberg, U. Brockmeier. *"Verifikation von ATHLET mit Hilfe der Einzeleffekttests ECTHOR und Patricia-SG2"*, RUB E-33, Mai 1993.
- [SCT-51] T. E. Schmidt. *"Der Wärmeübergang bei der Kondensation in Behältern und Rohren"*, Kältetechnik 1951, Nr. 11, S. 282 - 288.
- [SHM-79] M. M. Shah. *"A general correlation for heat transfer during film condensation inside pipes"*, Int. J. Heat Mass Transfer 22 (1979), S. 547 - 556.
- [SHM-81] M. M. Shah, *"Heat transfer during film condensation in tubes and annuli - a review of literature"*, American Society of Heating, Refrigerating and Air Conditioning 87 (1981), S. 1086 - 1105.
- [SIE-36] E. N. Sieder, G. E. Tate. *"Heat transfer and pressure drop of liquids in tubes"*, Industrial and Engineering Chemistry 28 (1936), Nr. 12, S. 1429 - 1435.

- [SIE-93] Siemens AG, Bereich Energieerzeugung (KWU). *"Konzeptbericht SWR600 - Ein fortschrittlicher Siedewasserreaktor mittlerer Leistung mit passiven Sicherheitsmerkmalen"*, Dezember 1993.
- [SIE-94] Siemens AG, Bereich Energieerzeugung (KWU). *"Konzeptbericht SWR1000 - Ein Siedewasserreaktor mittlerer Leistungsgröße mit passiven Sicherheitsmerkmalen"*, August 1994.
- [SOH-86] H. M. Soliman, *"The mist - annular transition during condensation and its influence on the heat transfer mechanism"*, Int. J. Multiphase Flow, Vol. 12 (1986), S. 277 - 288.
- [STF-89] F. Steinhoff. *"Thermo- und Fluidodynamikmodelle im Rechenprogramm DRUFAN und im Nachfolgeprogramm ATHLET zur Simulation von Separationsvorgängen und Gemischspiegelbewegungen in vertikalen Strömungskanälen - Teil1: Entwicklung der Modelle"*, GRS-A-1539, März 1989.
- [STK-88] K. Stephan. *"Wärmeübergang beim Kondensieren und beim Sieden"*, Springer-Verlag, Berlin Heidelberg New York London Paris Tokyo 1988.
- [TAT-82] T. N. Tandon, H. K. Varma, C. P. Gupta. *"A new flow regimes map for condensation inside horizontal tubes"*, Journal of Heat Transfer 104 (1982), S. 763 - 768.
- [TEV-88] V. Teschendorf, J. Miro, G. Lerchl. *"ATHLET ein fortschrittlicher Systemcode zur Analyse thermohydraulischer Prozesse"*. 12. GRS-Fachgespräch "Forschung zur Erhöhung der Reaktorsicherheit", Köln, 3.-4 November 1988.
- [TRD-78] D. P. Traviss, W. M. Rohsenow, A. B. Baron. *"Forced-Convection condensation inside tubes: A heat transfer equation for condenser design"*, American Society of Heating, Refrigerating and Airconditioning Engineers Transactions 79 (1978), Nr. 1, S. 157 - 165.
- [VDI-88] VDI-Wärmeatlas, Kapitel Ja1 - Ja29. *"Filmkondensation reiner Dämpfe"*, VDI-Verlag, Düsseldorf 1988.

- [WAG-69] G. B. Wallis. *"One dimensional two-phase flow"*, Mc Graw-Hill Book Company, New York 1969.
- [WAJ-74] J. v. d. Walt, D. G. Kröger. *"Heat transfer resistances during film condensation"*, 5th International Heat Transfer Conference, Tokyo 1974, Vol. 3, S. 284 - 288.



## **Anhang A: Listing von KONWAR**

```

      SUBROUTINE KONWAR(XM, EPS, GG, P, TD, TF, TW, DR, LA,
      *                MODEW, wert, LWRITE)
C=====
C Andreas Schaffrath
C Ruhr-Universitaet Bochum / Professur fuer Sicherheitsforschung und
C Reaktortechnik
C Forschungszentrum Juelich
C D-52425 Juelich
C Tel. 02461 / 613437
C 13.1.1995
C=====
CZ   Zweck: Berechnung der Kondensation in waagerechten Rohren
CZ-----
CF   Eingabeparameter:
CF
CF   Ausgabeparameter:
CF   MODEW   Mode (wird mit MODE ?? addiert)
CF   = x10   Spruehstroemung
CF   = x20   Uebergang zw. Sprueh- und Ringstroemung
CF   = x30   Ringstroemung
CF   = x40   Uebergang zw. Ring- und Schichtenstroemung
CF   = x50   Schichtenstroemung
CF   = x60   Blasenstroemung
CF   = x70   Schwallstroemung
CF   = x80   Pfropfenstroemung
CF   = x16   Uebergang Sprueh- und Blasenstroemung
CF   = x26   Uebergang Ring-Sprueh- und Blasenstroemung
CF   = x36   Uebergang Ring- und Blasenstroemung
CF   = x46   Uebergang Uebergang Schicht-Ring und Blasenstroemung
CF   = x47   Uebergang Uebergang Schicht-Ring und Schwallstroemung
CF   = x57   Uebergang Schichten- und Schwallstroemung
CF   = x58   Uebergang Schichten- und Pfropfenstroemung
CF
C-----
C   Vereinbarung der formalen Parameter
C-----
C
C   INCLUDE 'c'
C   INCLUDE 'ca'
C-----
C   Vereinbarung der formalen Parameter
C-----
C
C   DOUBLE PRECISION P, TD, TF, TW
C   INTEGER          MODEW
C   LOGICAL          LWRITE
C-----
CF   Verwendete Unterprogramme
CF
CF   BREBBER Blasen-, Pfropfen-, Schwallstroemung
CF   DAMLAM Laminare Dampfstroemung
CF   DAMTUR Turbulente Dampfstroemung
CF   DAMUEB Uebergangsdampfstroemung
CF   FLULAM Laminare Fluessigkeitsstroemung
CF   FLUTUR Turbulente Fluessigkeitsstroemung
CF   FLUUEB Uebergangsfluessigkeitsstroemung
CF   RING   Ringstroemung
CF   RINLAM Laminare Ringstroemung
CF   RINTUR Turbulente Ringstroemung
CF   RINUEB Uebergangsringstroemung
CF   SCHLAM Schichtenstroemung (laminar)
CF   SPRUEH Spruehstroemung
CF
CF   Stoffwertefunktionen
CF
CF   direkter Aufruf von:
CF   MPATS
CF   MPTAF
CF   MPAHFG
CF
CF   indirekte Aufrufe von:
CF   MPTSA
CF   S

```

```

CF  SNDS
CF  SPOLY
C-----
      DOUBLE PRECISION BREBBER, DAMLAM, DAMTUR, DAMUEB, FLULAM, FLUTUR,
      *                FLUEEB, RING, SCHLAM, SPRUEH,
      *                MPSAT, MPTAF, MPHPTL, MPHPTV
C-----
C  Vereinbarung der lokalen Groessen
C-----
CL  Lokale Variablen und Felder
C-----
      LOGICAL          dampf, zweipha, fluess, uebdampf, uebfluess
      DOUBLE PRECISION xm, jDstern, epsfl
      DOUBLE PRECISION REDM, REFM
      DOUBLE PRECISION DR, LA
      DOUBLE PRECISION RED, REF, REFIL, REFIT
      DOUBLE PRECISION ALPHAL, ALPHAT, EPS, MGES, GG
      DOUBLE PRECISION XAKTUELL
      DOUBLE PRECISION WERTT, WERTL, WERTRE, WERTLI, WERTLIL,
      *                WERT, WERTIN, WERTUEB
      DOUBLE PRECISION PI
C-----
CL  Vereinbarung der lokalen Variablen fuer die Stoffwerteberechnung
C-----
      DOUBLE PRECISION TS, TDW, TFW, CPD, CPDM, CPF, CPFM, DHV,
      *                ETAD, ETADM, ETAF, ETAFM, ETAFS, ETAFW,
      *                HD, HDS, HF, HFS, LAMD, LAMDM, LAMF, LAMFM,
      *                LAMFS, LAMFW, NUEFM, PRD, PRDM, PRF, PRFM,
      *                SIGMAM, ROD, RODM, ROF, ROFM
      DOUBLE PRECISION TDK, TFK, TDWK, TFWK, TSK
C-----
C-----Ausfuehrungsteil-----
C-----
C-----DATA-Anweisungen
      DATA PI /3.141592654/
C-----Initialisierung der Flags zur Kennzeichnung der vorhandenen Phasen
      dampf = .false.
      fluess = .false.
      zweipha = .false.
      uebdampf = .false.
      uebfluess = .false.
C-----
C-----Stoffwerteberechnung
C-----
      CALL PROP(P, TD, TF, TW, TS, TDW, TFW, CPD, CPDM, CPF, CPFM,
      *                DHV, ETAD, ETADM, ETAF, ETAFM, ETAFS, ETAFW, HD,
      *                HDS, HF, HFS, LAMD, LAMDM, LAMF, LAMFM, LAMFS,
      *                LAMFW, NUEFM, PRD, PRDM, PRF, PRFM, SIGMAM, ROD,
      *                RODM, ROF, ROFM, LWRITE)
C-----
C-----Temperaturumrechnung von C in K fuer TD, TF, TW, TFW, TDW
C-----
      TDK = TD + 273.15
      TWK = TW + 273.15
      TFK = TF + 273.15
      TFWK = TFW + 273.15
      TDWK = TDW + 273.15
      TSK = TS + 273.15
C-----
C-----Ausgabe der Eingabewerte
C-----
      IF (LWRITE) THEN
        WRITE (66, *) ' ++++++++'
        WRITE (66, *) ' EINGABEDATEN-EINGABEDATEN-EINGABEDATEN '
        WRITE (66, *) ' ++++++++'
        WRITE (66, 999) ' DR    = ', DR
        WRITE (66, 999) ' LAENGE HECU LA = ', LA
        WRITE (66, 999) ' xm    = ', xm
        WRITE (66, 999) ' GG    = ', GG
      ENDIF
      IF (EPS .LE. 1.0E-04) EPS = 1.0E-04
      IF (LWRITE) THEN

```

```

WRITE (66, 999) ' EPS = ', EPS
WRITE (66, *) ' ++++++'
WRITE (66, *) ' BERECHNUNG-BERECHNUNG-BERECHNUNG-BERECHNUNG '
WRITE (66, *) ' ++++++'
ENDIF

C-----Bestimmung der Reynoldszahlen RED, REDM, REF, REFM
C-----
RED = GG * xm * DR / ETAD
REDM = GG * xm * DR / ETADM
REF = GG * (1. - xm) * DR / ETAF
REFM = GG * (1. - xm) * DR / ETAFM
IF (LWRITE) THEN
  WRITE (66, *) ' ++++++'
  WRITE (66, *) ' reynoldszahlen-reynoldszahlen-reynoldszahlen'
  WRITE (66, *) ' ++++++'
  WRITE (66, 999) ' RED = ', RED, ' REDM = ', REDM
  WRITE (66, 999) ' REF = ', REF, ' REFM = ', REFM
  WRITE (66, *) ' ++++++'
  WRITE (66, *) ' DIMENSIONSLOSE DAMPFGESCHWINDIGKEIT '
  WRITE (66, *) ' ++++++'
ENDIF

C-----Bestimmung der dimensionslosen Dampfgeschwindigkeit jDstern
C-----
MGES = GG * PI / 4. * DR**2.
IF (MGES .LE. 1.0D-10) MGES = 1.0D-10
jDstern = 4.*xm*MGES/(PI*DR**2.*SQRT(9.81*DR*RODM*(ROFM-RODM)))
IF (jDstern .LE. 1.0E-4) jDstern = 1.0E-4
IF (LWRITE) THEN
  WRITE (66, 999) ' MGES = ', MGES, ' jDstern = ', jDstern
ENDIF

C-----Bestimmung des volumetrischen Fluessigkeitsanteils
C-----
epsfl = (1. - EPS)/EPS
IF (epsfl .GE. 1.0D+03) epsfl = 1.0D+03
IF (LWRITE) THEN
  WRITE (66, *) ' ++++++'
  WRITE (66, *) ' (1. - eps)/eps '
  WRITE (66, *) ' ++++++'
  WRITE (66, 999) ' epsfl = ', epsfl
  WRITE (66, *) ' ++++++'
  WRITE (66, *) ' BERECHNUNG DES ALPHAWERTES '
  WRITE (66, *) ' ++++++'
ENDIF

C-----
C Abfrage, welche Stroemung vorliegt (einphasig oder zweiphasig) und
C Belegung der entsprechenden Flags mit dem Wert true.
C-----
IF (xm .LE. 0.005) THEN
  IF (xm .LE. 0.0) THEN
C-----
    reine Fluessigkeitsstroemung
    fluess = .TRUE.
  ELSE
C-----
    Uebergangsgebiet Fluessigkeit-Zweiphasengebiet
    uebfluess = .TRUE.
  ENDIF
ELSEIF (xm .GE. 0.9995) THEN
  IF (xm .GE. 1.0) THEN
C-----
    reine Dampfstroemung
    dampf = .TRUE.
  ELSE
C-----
    Uebergangsgebiet Zweiphasen-Dampf
    uebdampf = .TRUE.
  ENDIF
ELSE
C-----
    Zweiphasenstroemung
    zweipha = .TRUE.
  ENDIF
ENDIF

C-----
C UEBERGANGSGEBIETE EIN- UND ZWEIPHASEN

```

```

C-----
      IF (uebdampf) THEN
        xaktuell = xm
        xm = 1.0
        dampf = .true.
        IF (LWRITE) THEN
          WRITE (66,*) '#####'
          WRITE (66,*) 'ALPHAWERT BEI XM = 1.0'
          WRITE (66,*) '#####'
        ENDIF
        GOTO 33
      ELSEIF (uebfluess) THEN
        xaktuell = xm
        xm = 0.0
        fluess = .true.
        IF (LWRITE) THEN
          WRITE (66,*) '#####'
          WRITE (66,*) 'ALPHAWERT BEI XM = 0.0'
          WRITE (66,*) '#####'
        ENDIF
        GOTO 44
      ENDIF
C-----
C-----
C  REINE DAMPFSTROEMUNG
C-----
C-----
33  IF (dampf) THEN
C-----
C
C  Laminare Dampfstroemung
C-----
C-----
      IF (redm .LE. 2300.) THEN
        IF (LWRITE) THEN
          WRITE (66,*) 'Laminare Dampfstroemung'
        ENDIF
        wertein = DAMLAM (DR, LAMDM, TDK, TWK)
        IF (LWRITE) THEN
          WRITE (66,999) '  DAMLAM = ', wertein
        ENDIF
C-----
C
C  Turbulente Dampfstroemung
C-----
C-----
      ELSEIF (redm .GE. 5000.) THEN
        IF (LWRITE) THEN
          WRITE (66,*) 'Turbulente Dampfstroemung'
        ENDIF
        wertein = DAMTUR (DR, LAMDM, PRDM, REDM, LWRITE)
        IF (LWRITE) THEN
          WRITE (66,999) '  DAMTUR = ', wertein
        ENDIF
C-----
C
C  Uebergangsgebiet
C-----
C-----
      ELSE
        IF (LWRITE) THEN
          WRITE (66,*) 'Reine Dampfstroemung im Uebergangsgebiet'
        ENDIF
        wertein = DAMUEB (DR, LAMDM, PRDM, REDM, TDK, TWK, LWRITE)
        IF (LWRITE) THEN
          WRITE (66,999) '  DAMUEB = ', wertein
        ENDIF
      ENDIF
      ENDIF
      IF (uebdampf) GOTO 333
C-----
C-----

```

```

C REINE FLUESSIGKEITSSTROEMUNG
C-----
C-----
44 IF (fluess) THEN
C-----
C
C Laminare Fluessigkeitsstroemung
C-----
      IF (refm .LE. 2300.) THEN
        IF (LWRITE) THEN
          WRITE (66,*) 'Laminare Fluessigkeitsstroemung'
        ENDIF
        wertein = FLULAM(DR, ETAF, ETAFW, LA, LAMF, PRF, REF)
        IF (LWRITE) THEN
          WRITE (66,999) ' FLULAM = ', WERTEIN
        ENDIF
C-----
C
C Turbulente Fluessigkeitsstroemung
C-----
      ELSEIF (refm .GE. 5000.) THEN
        IF (LWRITE) THEN
          WRITE (66,*) 'Turbulente Fluessigkeitsstroemung'
        ENDIF
        wertein = FLUTUR (DR, ETAF, ETAFW, LAMF, LAMFM, PRF,
          * PRFM, REF, REFM, LWRITE)
        IF (LWRITE) THEN
          WRITE (66,999) ' FLUTUR = ', WERTEIN
        ENDIF
C-----
C
C Uebergangsgebiet
C-----
      ELSE
        IF (LWRITE) THEN
          WRITE (66,*) 'Fluessigkeitsstroemung im Uebergangsgebiet'
        ENDIF
        wertein = FLUUEB (DR, ETAF, ETAFW, LA, LAMF, LAMFM, PRF,
          * PRFM, REF, REFM, LWRITE)
        IF (LWRITE) THEN
          WRITE (66,999) ' FLUUEB = ', WERTEIN
        ENDIF
      ENDIF
    ENDIF
C-----
C UEBERGANGSGEBIETE EIN - ZWEIPHASEN
C-----
      IF (uebfluess) GOTO 333
333 IF (uebdampf) THEN
      xm = 0.9995
      zweipha = .true.
      IF (LWRITE) THEN
        WRITE (66,*) '#####'
        WRITE (66,*) 'ALPHAWERT BEI XM = 0.9995'
        WRITE (66,*) '#####'
      ENDIF
      GOTO 55
    ELSEIF (uebfluess) THEN
      xm = 0.005
      zweipha = .true.
      IF (LWRITE) THEN
        WRITE (66,*) '#####'
        WRITE (66,*) 'ALPHAWERT BEI XM = 0.05'
        WRITE (66,*) '#####'
      ENDIF
      GOTO 55
    ENDIF
C-----
C-----

```

```

C  ZWEIPHASENSTROEMUNG
C -----
C
55  IF (zweipha) THEN
      IF (epsfl .LE. 0.5) THEN
C+++++
C Schichten-/Ring-/Spruehstroemung und Uebergangsbereiche
C      (0.001 =< )epsfl =< 0.5
C
C+++++
      IF (jDstern .LE. 0.4) THEN
C-----
C
C      wellige Schichtenstroemung
C      (0.001 =< ) jDstern =< 0.4
C
C-----
          IF (LWRITE) THEN
              WRITE (66, *) 'Schichtenstroemung'
          ENDIF
          MODEW = 50
          IF (LWRITE) THEN
              WRITE (66, '(A, I3)') ' MODEW = ', MODEW
          ENDIF
          wert = SCHLAM (DHV, DR, EPS, ETAFM, ETAFS, ETAFW, HD,
*                   HDS, HF, HFS, LAMFM, LAMFS, LAMFW, ROFM, TSK, TWK,
*                   LWRITE)
          IF (LWRITE) THEN
              WRITE (66, 999) ' SCHLAM = ', wert
          ENDIF
          ELSEIF (jDstern .GT. 0.4 .AND. jDstern .LT. 1.5) THEN
C-----
C      Uebergangsbereich zwischen Schichten- und Ringstroemung
C      0.4 < jDstern < 1.5
C-----
          IF (LWRITE) THEN
              WRITE (66, *) 'Uebergang zwischen Schichten und',
*                   ' Ringstroemung'
          ENDIF
          MODEW = 40
          IF (LWRITE) THEN
              WRITE (66, '(A, I3)') ' MODEW = ', MODEW
          ENDIF
C
C      U N T E R E   I N T E R P O L A T I O N S G R E N Z E   (schlam)
C
          wertli = SCHLAM (DHV, DR, EPS, ETAFM, ETAFS, ETAFW, HD,
*                   HDS, HF, HFS, LAMFM, LAMFS, LAMFW, ROFM, TSK, TWK,
*                   LWRITE)
          IF (LWRITE) THEN
              WRITE (66, *) 'schichten-schichten-schichten-schichten'
              WRITE (66, 999) ' WERTLI = ', wertli
          ENDIF
C
C      O B E R E   I N T E R P O L A T I O N S G R E N Z E   (rin...)
C
          wertre = RING (CPF, DHV, DR, ETAFM, ETAFW, ETAFS, HD,
*                   HDS, HF, HFS, LAMFM, LAMFS, LAMFW, MGES,
*                   NUEFM, P, PRF, PRFM, RODM, ROFM, RED,
*                   ROD, ROF, LA, SIGMAM, TSK, TWK, XM,
*                   REFM, REF, LWRITE)
          IF (LWRITE) THEN
              WRITE (66, *) 'ring-ring-ring-ring-ring-ring-ring-ring'
              WRITE (66, 999) ' WERTRE = ', wertre
          ENDIF
C
C      Interpolation wert
C
          wert = wertli + (jDstern - 0.4) / (1.5 - 0.4) * (wertre - wertli)

```

```

      IF (LWRITE) THEN
        WRITE (66,*) 'INTERPOLATION-INTERPOLATION-INTERPOLATION'
        WRITE (66,999) ' WERT = ', wert
      ENDIF
    ELSEIF (jDstern .GE. 1.5 .AND. jDstern .LE. 6.0) THEN
C-----
C
C   Ringstroemung (mit Tropfen im Kern)
C       1.5 =< jDstern =< 6.0
C-----
      IF (LWRITE) THEN
        WRITE (66, *) 'Ringstroemung'
      ENDIF
      MODEW = 30
      IF (LWRITE) THEN
        WRITE (66, '(A, I3)') ' MODEW = ', MODEW
      ENDIF
      wert = RING (CPF, DHV, DR, ETAFM, ETAFW, ETAFS, HD,
*               HDS, HF, HFS, LAMFM, LAMFS, LAMFW, MGES,
*               NUEFM, P, PRF, PRFM, RODM, ROFM, RED,
*               ROD, ROF, LA, SIGMAM, TSK, TWK, XM,
*               REFM, REF, LWRITE)
      IF (LWRITE) THEN
        WRITE (66,999) ' RING = ', wert
      ENDIF
C
    ELSEIF (jDstern .GT. 6.0 .AND. jDstern .LT. 6.5) THEN
C-----
C
C   Uebergangsbereich zwischen Ring- und Spruehstroemung
C       6.0 < jDstern < 6.5
C-----
      IF (LWRITE) THEN
        WRITE (66, *) 'Uebergang zwischen Sprueh- und',
*               'Ringstroemung'
      ENDIF
      MODEW = 20
      IF (LWRITE) THEN
        WRITE (66, '(A, I3)') ' MODEW = ', MODEW
      ENDIF
C
C   U N T E R E   I N T E R P O L A T I O N S G R E N Z E   (rin...)
C
      wertli = RING (CPF, DHV, DR, ETAFM, ETAFW, ETAFS, HD,
*               HDS, HF, HFS, LAMFM, LAMFS, LAMFW, MGES,
*               NUEFM, P, PRF, PRFM, RODM, ROFM, RED,
*               ROD, ROF, LA, SIGMAM, TSK, TWK, XM,
*               REFM, REF, LWRITE)
      IF (LWRITE) THEN
        WRITE (66,*) 'ring-ring-ring-ring-ring-ring-ring-ring'
        WRITE (66,999) ' WERTLI = ', wertli
      ENDIF
C
C   O B E R E   I N T E R P O L A T I O N S G R E N Z E   (sprueh)
C
      IF (LWRITE) THEN
        WRITE (66,*) 'sprueh-sprueh-sprueh-sprueh'
      ENDIF
      wertre = SPRUEH (CPD, DHV, DR, ETAD, ETAF, GG, LAMF,
*               PRD, TSK, TWK, XM, LWRITE)
      IF (LWRITE) THEN
        WRITE (66,999) ' WERTRE = ', wertre
      ENDIF
C
C   Interpolation wert
C
      wert = wertli+(jDstern-6.0)/(6.5-6.0)*(wertre-wertli)
      IF (LWRITE) THEN
        WRITE (66,*) 'INTERPOLATION-INTERPOLATION-INTERPOLATION'
        WRITE (66,999) ' WERT = ', wert
      ENDIF

```



```

        ENDIF
      ELSE
C-----
C
C   Spruehstroemung
C       6.5 =< jDstern (=< 20.)
C-----
C
C       IF (LWRITE) THEN
C           WRITE (66, *) 'Spruehstroemung'
C       ENDIF
C       MODEW = 10
C       IF (LWRITE) THEN
C           WRITE (66, '(A, I3)') ' MODEW = ', MODEW
C       ENDIF
C       wert = SPRUEH (CPD, DHV, DR, ETAD, ETAF, GG, LAMF,
C           *           PRD, TSK, TWK, XM, LWRITE)
C       IF (LWRITE) THEN
C           WRITE (66, 999) 'SPRUEH   = ', wert
C       ENDIF
C   ELSE IF (epsfl .GT. 0.5 .AND. epsfl .LT. 1.5) THEN
C+++++++
C   Uebergangsbereiche
C       0.5 < epsfl < 1.5
C+++++++
C       IF (jDstern .LE. 0.01) THEN
C-----
C
C   Uebergang zwischen welliger Schichtenstroemung
C   und Pfropfenstroemung
C       (0.001 =<) jDstern =< 0.01
C-----
C
C       IF (LWRITE) THEN
C           WRITE (66, *) 'Uebergang zwischen Schichten- und Pfropfen
C           *           stroemung'
C       ENDIF
C       MODEW = 58
C       IF (LWRITE) THEN
C           WRITE (66, '(A, I3)') ' MODEW = ', MODEW
C       ENDIF
C
C   L I N K E   I N T E R P O L A T I O N S G R E N Z E   (schlam)
C
C       wertli = SCHLAM (DHV, DR, EPS, ETAFM, ETAFS, ETAFW, HD,
C           *           HDS, HF, HFS, LAMFM, LAMFS, LAMFW, ROFM, TSK, TWK,
C           *           LWRITE)
C       IF (LWRITE) THEN
C           WRITE (66, *) 'schichten-schichten-schichten-schichten'
C           WRITE (66, 999) ' WERTLI = ', wertli
C       ENDIF
C
C   R E C H T E   I N T E R P O L A T I O N S G R E N Z E   (brebber)
C
C       IF (LWRITE) THEN
C           WRITE (66, *) 'pfropfen-pfropfen-pfropfen-pfropfen'
C       ENDIF
C       wertre = BREBBER (DR, ETAF, ETAFW, LA, LAMF, PRF,
C           *           REF, REDM, ROD, ROF, XM, P, LWRITE)
C       IF (LWRITE) THEN
C           WRITE (66, 999) ' WERTRE = ', wertre
C       ENDIF
C
C       Interpolation wert
C
C       wert = wertli + (epsfl-0.5) / (1.5-0.5)*(wertre- wertli)
C       IF (LWRITE) THEN
C           WRITE (66, *) 'INTERPOLATION-INTERPOLATION-INTERPOLATION'
C           WRITE (66, 999) ' WERT = ', wert

```

```

      ENDIF
      ELSEIF (jDstern .GT. 0.01 .AND. jDstern .LT. 0.4) THEN
C-----
C
C   Uebergang zwischen welliger Schichtenstroemung
C   und Schwallstroemung
C       0.01 < jDstern < 0.4
C-----
      IF (LWRITE) THEN
        WRITE (66, *) 'Uebergang zwischen Schichten und',
          *           'Schwallstroemung'
      ENDIF
      MODEW = 57
      IF (LWRITE) THEN
        WRITE (66, '(A, I3)') ' MODEW = ', MODEW
      ENDIF
C
C   L I N K E   I N T E R P O L A T I O N S G R E N Z E   (schlam)
C
      wertli = SCHLAM (DHV, DR, EPS, ETAFM, ETAFS, ETAFW, HD,
        *           HDS, HF, HFS, LAMFM, LAMFS, LAMFW, ROFM, TSK, TWK,
        *           LWRITE)
      IF (LWRITE) THEN
        WRITE (66, *) 'schichten-schichten-schichten-schichten'
        WRITE (66, 999) ' WERTLI = ', wertli
      ENDIF
C
C   R E C H T E   I N T E R P O L A T I O N S G R E N Z E   (brebber)
C
      IF (LWRITE) THEN
        WRITE (66, *) 'schwall-schwall-schwall-schwall-schwall'
      ENDIF
      wertre = BREBBER (DR, ETAF, ETAFW, LA, LAMF, PRF,
        *           REF, REDM, ROD, ROF, XM, P, LWRITE)
      IF (LWRITE) THEN
        WRITE (66, 999) ' WERTRE = ', wertre
      ENDIF
C
C   Interpolation wert
C
      wert = wertli + (epsfl-0.5) / (1.5-0.5)*(wertre- wertli)
      IF (LWRITE) THEN
        WRITE (66, *) 'INTERPOLATION-INTERPOLATION-INTERPOLATION'
        WRITE (66, 999) ' WERT = ', wert
      ENDIF
      ELSEIF (jDstern .GE. 0.4 .AND. jDstern .LE. 0.5) THEN
C-----
C
C   Uebergangsbereich zwischen dem Uebergangsbereich
C   zwischen Schichten- und Ringstroemung und Schwall-
C   stroemung
C       0.4 =< jDstern =< 0.5
C-----
      IF (LWRITE) THEN
        WRITE (66, *) 'Uebergang Uebergang Schicht-Ring',
          *           ' und Schwallstr.'
      ENDIF
      MODEW = 47
      IF (LWRITE) THEN
        WRITE (66, '(A, I3)') ' MODEW = ', MODEW
      ENDIF
C
C   L I N K E   I N T E R P O L A T I O N S G R E N Z E   (schlam/ring)
C
C   U N T E R E   I N T E R P O L A T I O N S G R E N Z E   (schlam)
C
      wertli = SCHLAM (DHV, DR, EPS, ETAFM, ETAFS, ETAFW, HD,
        *           HDS, HF, HFS, LAMFM, LAMFS, LAMFW, ROFM, TSK, TWK,
        *           LWRITE)
      IF (LWRITE) THEN

```

```

        WRITE (66,*) 'schichten-schichten-schichten-schichten'
        WRITE (66,999) ' WERTLI = ', wertli
    ENDIF
C
C      O B E R E   I N T E R P O L A T I O N S G R E N Z E   (rin...)
C
        wertre = RING (CPF, DHV, DR, ETAFM, ETAFW, ETAFS, HD,
*                   HDS, HF, HFS, LAMFM, LAMFS, LAMFW, MGES,
*                   NUEFM, P, PRF, PRFM, RODM, ROFM, RED,
*                   ROD, ROF, LA, SIGMAM, TSK, TWK, XM,
*                   REFM, REF, LWRITE)
        IF (LWRITE) THEN
            WRITE (66,*) 'ring-ring-ring-ring-ring-ring-ring-ring'
            WRITE (66,999) ' WERTRE = ', wertre
        ENDIF
C
C      Interpolation wert
C
        wertlil = wertli+(jDstern-0.4)/(1.5-0.4)*(wertre- wertli)
        IF (LWRITE) THEN
            WRITE (66,*) 'INTERPOLATION-INTERPOLATION-INTERPOLATION'
            WRITE (66,999) ' WERT = ', wertlil
        ENDIF
C
C      R E C H T E   I N T E R P O L A T I O N S G R E N Z E   (brebber)
C
        IF (LWRITE) THEN
            WRITE (66,*) 'schwall-schwall-schwall-schwall-schwall'
        ENDIF
        wertre = BREBBER (DR, ETAF, ETAFW, LA, LAMF, PRF,
*                   REF, REDM, ROD, ROF, XM, P, LWRITE)
C
C      Interpolation wert
C
        wert = wertlil+(epsfl-0.5)/(1.5-0.5)*(wertre-wertlil)
        IF (LWRITE) THEN
            WRITE (66,*) 'INTERPOLATION-INTERPOLATION-INTERPOLATION'
            WRITE (66,999) ' WERT = ', wert
        ENDIF
        ELSEIF (jDstern .GT. 0.5 .AND. jDstern .LT. 1.5) THEN
C-----
C
C      Uebergangsbereich zwischen dem Uebergangsbereich
C      zwischen Schichten- und Ringstroemung und Blasen-
C      stroemung
C      0.5 < jDstern < 1.5
C-----
        IF (LWRITE) THEN
            WRITE (66, *) 'Uebergang Uebergang Schicht-Ring und',
*                   ' Blasenstroemung'
        ENDIF
        MODEW = 46
        IF (LWRITE) THEN
            WRITE (66, '(A, I3)') ' MODEW = ', MODEW
        ENDIF
C
C      L I N K E   I N T E R P O L A T I O N S G R E N Z E   (schlam/ring)
C
C      U N T E R E   I N T E R P O L A T I O N S G R E N Z E   (schlam)
C
        wertli = SCHLAM (DHV, DR, EPS, ETAFM, ETAFS, ETAFW, HD,
*                   HDS, HF, HFS, LAMFM, LAMFS, LAMFW, ROFM, TSK, TWK,
*                   LWRITE)
        IF (LWRITE) THEN
            WRITE (66,*) 'schichten-schichten-schichten-schichten'
            WRITE (66,999) ' WERTLI = ', wertli
        ENDIF
C
C      O B E R E   I N T E R P O L A T I O N S G R E N Z E   (ring)
C

```

```

wertre = RING (CPF, DHV, DR, ETAFM, ETAFW, ETAFS, HD,
*           HDS, HF, HFS, LAMFM, LAMFS, LAMFW, MGES,
*           NUEFM, P, PRF, PRFM, RODM, ROFM, RED,
*           ROD, ROF, LA, SIGMAM, TSK, TWK, XM,
*           REFM, REF, LWRITE)
IF (LWRITE) THEN
  WRITE (66,*) 'ring-ring-ring-ring-ring-ring-ring-ring'
  WRITE (66,999) ' WERTRE = ', wertre
ENDIF

C
C   Interpolation wert
C

wertlil = wertli+(jDstern-0.4)/(1.5-0.4)*(wertre-wertli)
IF (LWRITE) THEN
  WRITE (66,*) 'INTERPOLATION-INTERPOLATION-INTERPOLATION'
  WRITE (66,999) ' WERT = ', wertlil
ENDIF

C
C   R E C H T E   I N T E R P O L A T I O N S G R E N Z E   (brebber)
C

IF (LWRITE) THEN
  WRITE (66,*) 'blasen-blasen-blasen-blasen-blasen'
ENDIF
wertre = BREBBER (DR, ETAF, ETAFW, LA, LAMF, PRF,
*           REF, REDM, ROD, ROF, XM, P, LWRITE)
IF (LWRITE) THEN
  WRITE (66,999) ' WERTRE = ', wertre
ENDIF

C
C   Interpolation wert
C

wert = wertlil+(epsfl-0.5)/(1.5-0.5)*(wertre-wertlil)
IF (LWRITE) THEN
  WRITE (66,*) 'INTERPOLATION-INTERPOLATION-INTERPOLATION'
  WRITE (66,999) ' WERT = ', wert
ENDIF
ELSEIF (jDstern .GE. 1.5 .AND. jDstern .LE. 6.0) THEN
-----
C
C   Uebergangsbereich zwischen Ringstroemung (mit
C   Tropfen im Kern) und Blasenstroemung
C   1.5 =< jDstern =< 6.0
C
-----
IF (LWRITE) THEN
  WRITE (66,*) 'Uebergang zwischen Ring- und Blasen-',
*           ' stroemung'
ENDIF
MODEW = 36
IF (LWRITE) THEN
  WRITE (66, '(A, I3)') ' MODEW = ', MODEW
ENDIF

C
C   L I N K E   I N T E R P O L A T I O N S G R E N Z E   (ring)
C

wertli = RING (CPF, DHV, DR, ETAFM, ETAFW, ETAFS, HD,
*           HDS, HF, HFS, LAMFM, LAMFS, LAMFW, MGES,
*           NUEFM, P, PRF, PRFM, RODM, ROFM, RED,
*           ROD, ROF, LA, SIGMAM, TSK, TWK, XM,
*           REFM, REF, LWRITE)
IF (LWRITE) THEN
  WRITE (66,*) 'ring-ring-ring-ring-ring-ring-ring-ring'
  WRITE (66,999) ' WERTLI = ', wertli
ENDIF

C
C   R E C H T E   I N T E R P O L A T I O N S G R E N Z E   (brebber)
C

IF (LWRITE) THEN
  WRITE (66,*) 'blasen-blasen-blasen-blasen-blasen'
ENDIF
wertre = BREBBER (DR, ETAF, ETAFW, LA, LAMF, PRF,
*           REF, REDM, ROD, ROF, XM, P, LWRITE)

```

```

      IF (LWRITE) THEN
        WRITE (66,999) ' WERTRE = ', wertre
      ENDIF

C
C      Interpolation wert
C
      wert = wertli+(epsfl-0.5)/(1.5-0.5)*(wertre-wertli)
      IF (LWRITE) THEN
        WRITE (66,*) 'INTERPOLATION-INTERPOLATION-INTERPOLATION'
        WRITE (66,999) ' WERT = ', wert
      ENDIF
      ELSEIF (jDstern .GT. 6.0 .AND. jDstern .LT. 6.5) THEN
C-----
C
C      Uebergangsbereich zwischen dem Uebergangsbereich
C      zwischen Ring- und Spruehstroemung und Blasen-
C      stroemung
C      6.0 < jDstern < 6.5
C-----
C
      IF (LWRITE) THEN
        WRITE (66, *) 'Uebergang Uebergang Ring-Sprueh und',
          *           ' Blasenstroemung'
      ENDIF
      MODEW = 26
      IF (LWRITE) THEN
        WRITE (66,'(A, I3)') ' MODEW = ', MODEW
      ENDIF

C
C      L I N K E   I N T E R P O L A T I O N S G R E N Z E   (sprueh/ring)
C
C
C      U N T E R E   I N T E R P O L A T I O N S G R E N Z E   (ring)
C
      wertli = RING (CPF, DHV, DR, ETAFM, ETAFW, ETAFS, HD,
        *           HDS, HF, HFS, LAMFM, LAMFS, LAMFW, MGES,
        *           NUEFM, P, PRF, PRFM, RODM, ROFM, RED,
        *           ROD, ROF, LA, SIGMAM, TSK, TWK, XM,
        *           REFM, REF, LWRITE)
      IF (LWRITE) THEN
        WRITE (66,*) 'ring-ring-ring-ring-ring-ring-ring-ring'
        WRITE (66,999) ' WERTLI = ', wertli
C
C      O B E R E   I N T E R P O L A T I O N S G R E N Z E   (sprueh)
C
      WRITE (66,*) 'sprueh-sprueh-sprueh-sprueh'
      ENDIF
      wertre = SPRUEH (CPD, DHV, DR, ETAD, ETAF, GG, LAMF,
        *           PRD, TSK, TWK, XM, LWRITE)
      IF (LWRITE) THEN
        WRITE (66,999) ' WERTRE = ', wertre
      ENDIF

C
C      Interpolation wert
C
      wertlil = wertli+(jDstern-6.0)/(6.5-6.0)*(wertre-wertli)
      IF (LWRITE) THEN
        WRITE (66,*) 'INTERPOLATION-INTERPOLATION-INTERPOLATION'
        WRITE (66,999) ' WERT = ', wertlil
      ENDIF

C
C      R E C H T E   I N T E R P O L A T I O N S G R E N Z E   (brebber)
C
      IF (LWRITE) THEN
        WRITE (66,*) 'blasen-blasen-blasen-blasen-blasen'
      ENDIF
      wertre = BREBBER (DR, ETAF, ETAFW, LA, LAMF, PRF,
        *           REF, REDM, ROD, ROF, XM, P, LWRITE)
      IF (LWRITE) THEN
        WRITE (66,999) ' WERTRE = ', wertre
      ENDIF
C

```

```

C          Interpolation wert
C
      wert = wertli+(epsfl-0.5)/(1.5-0.5)*(wertre-wertli)
      IF (LWRITE) THEN
        WRITE (66,*) 'INTERPOLATION-INTERPOLATION-INTERPOLATION'
        WRITE (66,999) ' WERT = ', wert
      ENDIF
    ELSE
C-----
C
C      Uebergangsbereich zwischen Spruehstroemung und
C      Blasenstroemung
C      6.5 =< jDstern (= < 20.)
C-----
      IF (LWRITE) THEN
        WRITE (66, *) 'Uebergang zwischen Sprueh- und',
          *           ' Blasenstroemung'
      ENDIF
      MODEW = 16
      IF (LWRITE) THEN
        WRITE (66,'(A, I3)') ' MODEW = ', MODEW
C
C      L I N K E   I N T E R P O L A T I O N S G R E N Z E   (sprueh)
C
        WRITE (66,*) 'sprueh-sprueh-sprueh-sprueh'
      ENDIF
      wertli = SPRUEH (CPD, DHV, DR, ETAD, ETAF, GG, LAMF,
        *           PRD, TSK, TWK, XM, LWRITE)
      IF (LWRITE) THEN
        WRITE (66,999) ' WERTLI = ', wertli
      ENDIF
C
C      R E C H T E   I N T E R P O L A T I O N S G R E N Z E   (briebber)
C
      IF (LWRITE) THEN
        WRITE (66,*) 'blasen-blasen-blasen-blasen-blasen'
      ENDIF
      wertre = BREBBER (DR, ETAF, ETAFW, LA, LAMF, PRF,
        *           REF, REDM, ROD, ROF, XM, P, LWRITE)
      IF (LWRITE) THEN
        WRITE (66,999) ' WERTRE = ', wertre
      ENDIF
C
C      Interpolation wert
C
      wert = wertli+(epsfl-0.5)/(1.5-0.5)*(wertre- wertli)
      IF (LWRITE) THEN
        WRITE (66,*) 'INTERPOLATION-INTERPOLATION-INTERPOLATION'
        WRITE (66,999) ' WERT = ', wert
      ENDIF
    ELSE
C+++++
C
C      Pfropfen-/Schwall-/Blasenstroemung
C      1.5 =< epsfl(< 10.)
C+++++
      IF (jDstern .LE. 0.01) THEN
C-----
C
C      Pfropfenstroemung
C      (0.001 =<) jDstern =< 0.01
C-----
      IF (LWRITE) THEN
        WRITE (66, *) 'Pfropfenstroemung'
      ENDIF
      MODEW = 80
      IF (LWRITE) THEN
        WRITE (66,'(A, I3)') ' MODEW = ', MODEW

```

```

        ENDIF
        GOTO 77
    ELSEIF (jDstern .GT. 0.01 .AND. jDstern .LT. 0.5) THEN
C-----
C
C   Schwallstroemung
C       0.01 < jDstern < 0.5
C-----
        IF (LWRITE) THEN
            WRITE (66, *) 'Schwallstroemung'
        ENDIF
        MODEW = 70
        IF (LWRITE) THEN
            WRITE (66, '(A, I3)') ' MODEW = ', MODEW
        ENDIF
        GOTO77
    ELSE
C-----
C
C   Blasenstroemung
C       0.5 =< jDstern (<= 20.)
C-----
        IF (LWRITE) THEN
            WRITE (66, *) 'Blasenstroemung'
        ENDIF
        MODEW = 60
        IF (LWRITE) THEN
            WRITE (66, '(A, I3)') ' MODEW = ', MODEW
        ENDIF
        GOTO77
    ENDIF
ENDIF
GOTO 111
C-----
C Kopie um Sprung in IF Block zu umgehen
C-----
77  CONTINUE
    wert = BREBBER (DR, ETAF, ETAFW, LA, LAMF, PRF,
        *           REF, REDM, ROD, ROF, XM, P, LWRITE)
        IF (LWRITE) THEN
            WRITE (66, 999) ' BREBBER = ', wert
        ENDIF
111  CONTINUE
999  FORMAT (2(A, D15.8, TR6))
99   FORMAT (3(A, D15.8, TR6))
C-----
C  UEBERGANGSGEBIETE EIN- ZWEIFHASEN
C-----
    IF (dampf .OR. fluess) THEN
        wert = wertein
    ELSEIF (uebdampf) THEN
        wertueb = wert + (xaktuell - 0.9995) *
        *       (wertein-wert)/(1.0-0.9995)
        IF (LWRITE) THEN
            WRITE (66, *) '#####'
            WRITE (66, 999) ' xaktuell = ', xaktuell
            WRITE (66, 999) ' wertein = ', wertein, ' wert = ', wert
            WRITE (66, 999) ' WERT = ', wertueb
        ENDIF
        wert = wertueb
    ELSEIF (uebfluess) THEN
        wertueb = wertein + (xaktuell) *
        *       (wert-wertein)/(0.005-0.0)
        IF (LWRITE) THEN
            WRITE (66, *) '#####'
            WRITE (66, 999) ' xaktuell = ', xaktuell
            WRITE (66, 999) ' wertein = ', wertein, ' wert = ', wert
            WRITE (66, 999) ' WERT = ', wertueb
        ENDIF
    ENDIF

```

```
    wert = wertueb  
ENDIF  
END
```



## **Anhang B: Listing der Funktionen zur Berechnung der Wärmeübergangskoeffizienten**

BREBER .....	B- 2
DAMLAM .....	B- 4
DAMTUR .....	B- 5
DAMUEB .....	B- 6
FLUTUR .....	B- 8
FLUUEB .....	B- 9
RING .....	B-10
RINLAM .....	B-12
RINTUR .....	B-14
RINUEB .....	B-16
SPRUEH .....	B-17
SCHLAM .....	B-18

## BREBER

```

      FUNCTION BREBBER (DR, ETAF, ETAFW, LA, LAMF, PRF, REF, REDM,
      *                      ROD, ROF, XM, P, LWRITE)
C=====
C Andreas Schaffrath
C Ruhr-Universitaet Bochum / Professur fuer Sicherheitsforschung und
C Reaktortechnik
C Forschungszentrum Juelich
C D-52425 Juelich
C Tel. 02461 / 613437
C 23.1.1995
C=====
C Berechnung des Waermeuebergangskoeffizienten bei turbulenter
C Fluessigkeitsstroemung nach Dittus-Boelter bzw. Sieder-Tate
C=====
C Variablen:
C ALPHA Waermeuebergangskoeffizient (W/m2K)
C ALPHASL Waermeuebergangskoeffizient nach Sieder-Tate laminar (W/m2K)
C ALPHAST Waermeuebergangskoeffizient nach Sieder-Tate turbulent (W/m2K)
C BREBBER Waermeuebergangskoeffizient nach Brebber (W/m2K)
C CD Konstante fuer Druckverlustbestimmung GE 402.4 S. 14
C CF Konstante fuer Druckverlustbestimmung GE 402.4 S. 14
C DR Durchmesser des Rohres (m)
C ETAF dynamische Viskositaet der Fluessigkeit (kg/sm)
C ETAFW dynamische Viskositaet der Fluessigkeit bei der Wandtempera-
C tur (kg/sm)
C LA Laenge bis zur aktuellen Stelle im Rohr (-)
C LAMF Waermeleitfaehigkeit der Fluessigkeit bei (W/mK)
C M Exponent in Brebber Korrelation ( M = 0.45)
C N Konstante fuer Druckverlustbestimmung GE 402.4 S. 14
C ND Konstante fuer Druckverlustbestimmung GE 402.4 S. 14
C NF Konstante fuer Druckverlustbestimmung GE 402.4 S. 14
C P Druck (Pa)
C PHI2 Zweiphasenmultiplikator
C PRF Prandtl-Zahl der Fluessigkeit (-)
C REF Reynolds-Zahl des Fluessigkeit (-)
C REDM Reynolds-Zahl des Dampfes bei 0.5(TW + TS) (-)
C X Wurzel Martinelli-Nelson Parameter
C X2 Martinelli-Nelson Parameter
C=====
      DOUBLE PRECISION ALPHA, ALPHASL, ALPHAST, BREBBER, CD, CF, DR,
      * ETAF, ETAFW, LA, LAMF, M, N, ND, NF, PHI2, PRF,
      * REF, X, X2, REDM, ROF, ROD, P, XM
      LOGICAL LWRITE
      DATA M /0.45/
      ALPHASL = 0.0
      ALPHAST = 0.0
C-----Fallunterscheidung REF laminar oder turbulent
      IF (REF .GE. 2000.) THEN
C
C-----Sieder-Tate turbulent
C
      ALPHAST = 0.023 * LAMF / DR * REF ** 0.8 * PRF ** 0.33 * (ETAF /
      * ETAFW) ** 0.14
      ELSE
C
C-----Sieder-Tate laminar
C
      ALPHASL = 1.86 * LAMF / DR * (REF * PRF * DR / LA) ** 0.33
      * (ETAF / ETAFW) ** 0.14
      ENDIF
      ALPHA = ALPHASL + ALPHAST
C
C-----Bestimmung des Martinelli-Nelson Parametres
C
      IF (REDM .GE. 2000.) THEN
        ND = 0.25
        CD = 0.079

```

```

      ELSE
        ND = 1.0
        CD = 16.0
      ENDIF
      IF (REF .GE. 2000.) THEN
        NF = 0.25
        CF = 0.079
      ELSE
        NF = 1.0
        CF = 16.0
      ENDIF
      IF (REF .GE. 1000. .AND. REDM .GE. 1000.) THEN
        IF (P .LE. 1.1D+06) THEN
          N = 5.13
        ELSE
          N = 4.0
        ENDIF
      ELSEIF (REF .LT. 1000. .AND. REDM .LT. 1000.) THEN
        N = 2.75
      ELSE
        N = 3.5
      ENDIF
      X2 = (REDM**ND / REF**NF) * CF / CD * ROD / ROF *
*      ((1. - XM) / XM)**2.
      X = sqrt (X2)
      PHIF2 = (1. + (1. / X) ** (2. / N)) ** N
C-----WUEK nach BREBBER
      BREBBER = ALPHA * PHIF2 ** M
      IF (LWRITE) THEN
        WRITE (66,*) '-----'
        WRITE (66,*) ' HILFSGROESSEN:'
        WRITE (66,999) ' ALPHAST = ', ALPHAST, ' ALPHASL = ', ALPHASL
        WRITE (66,999) ' N      = ', N, ' M      = ', M
        WRITE (66,999) ' ND     = ', ND, ' NF     = ', NF
        WRITE (66,999) ' CF     = ', CF, ' CD     = ', CD
        WRITE (66,999) ' X2     = ', X2, ' X      = ', X
        WRITE (66, ' (A, D15.8)') ' PHIF2 = ', PHIF2
999  FORMAT (2(A, D15.8, TR6))
        WRITE (66,*) '-----'
      ENDIF
    END

```

## DAMLAM

```
FUNCTION DAMLAM (DR, LAMDM, TDK, TWK)
C=====
C Andreas Schaffrath
C Ruhr-Universitaet Bochum / Professur fuer Sicherheitsforschung und
C Reaktortechnik
C Forschungszentrum Juelich
C D-52425 Juelich
C Tel. 02461 / 613437
C 23.1.1995
C=====
C Berechnung des Waermeuebergangskoeffizienten bei laminaren Dampf-
C keitsstroemung nach Hausen
C=====
C Variablen:
C ALPHA   Waermeuebergangskoeffizient (W/m2K)
C DAMLAM  WUEK bei laminarer Dampfstroemung (W/m2K)
C DR      Durchmesser des Rohres (m)
C LAMDM   Waermeleitfaehigkeit des Dampfes bei (TWK+TDK)*0.5 (W/mK)
C TDK     Dampftemperatur (K)
C TWK     Wandtemperatur (K)
C=====
      DOUBLE PRECISION ALPHA, DAMLAM, DR, LAMDM, TDK, TWK
      ALPHA = 3.66 * LAMDM / DR * (TDK / TWK)**0.25
      DAMLAM = ALPHA
      END
```

## DAMTUR

```

      FUNCTION DAMTUR (DR, LAMDM, PRDM, REDM, LWRITE)
C=====
C Andreas Schaffrath
C Ruhr-Universitaet Bochum / Professur fuer Sicherheitsforschung und
C Reaktortechnik
C Forschungszentrum Juelich
C D-52425 Juelich
C Tel. 02461 / 613437
C 23.1.1995
C=====
C Berechnung des Waermeuebergangskoeffizienten bei turbulenter Dampf-
C stroemung nach Dittus-Boelter
C=====
C Variablen:
C ALPHA   Waermeuebergangskoeffizient (W/m2K)
C DAMTUR  WUEK bei laminarer Dampfstroemung (W/m2K)
C DR       Durchmesser des Rohres (m)
C LAMDM    Waermeleitfaehigkeit des Dampfes bei (TW+TD)*0.5 (W/mK)
C PRDM     Prandtl-Zahl des Dampfes bei (TW+TD)*0.5 (-)
C REDM     Reynolds-Zahl des Dampfes bei (TW+TD)*0.5 (-)
C=====
      DOUBLE PRECISION ALPHA, DR, DAMTUR, LAMDM, PRDM, REDM
      LOGICAL LWRITE
C      IF (REDM .GT. 7.00D+04) THEN
C        REDM = 7.00D+04
C        IF (LWRITE) WRITE (66,'(A)') 'Reynolds-Zahl groesser 70000'
C      ENDIF
C      IF (PRDM .LT. 0.7) THEN
C        PRDM = 0.7
C        IF (LWRITE) WRITE (66,'(A)') 'Prandtl-Zahl kleiner 0.7'
C      ELSEIF (PRDM .GT. 1.00D+02) THEN
C        PRDM = 1.00D+02
C        IF (LWRITE) WRITE (66,'(A)') 'Prandtl-Zahl groesser 100'
C      ENDIF
      ALPHA = 0.023 * LAMDM / DR * REDM**0.8 * PRDM**0.4
      DAMTUR = ALPHA
      END

```

## DAMUEB

```

      FUNCTION DAMUEB (DR, LAMDM, PRDM, REDM, TDK, TWK, LWRITE)
C=====
C Andreas Schaffrath
C Ruhr-Universitaet Bochum / Professur fuer Sicherheitsforschung und
C Reaktortechnik
C Forschungszentrum Juelich
C D-52425 Juelich
C Tel. 02461 / 613437
C 12.1.1995
C=====
C Berechnung des Waermeuebergangskoeffizienten im Uebergangsbereich
C zwischen laminarer und turbulenter Dampfstroemung
C=====
C Variablen:
C ALPHA   Waermeuebergangskoeffizient (W/m2K)
C ALPHAL  Waermeuebergangskoeffizient laminar (W/m2K)
C ALPHAT  Waermeuebergangskoeffizient turbulent (W/m2K)
C DAMUEB  WUEK bei Dampfstroemung im Uebergangsgebiet (W/m2K)
C DR      Durchmesser des Rohres (m)
C LAMDM   Waermeleitfaehigkeit des Dampfes bei (TWK+TDK)*0.5 (W/mK)
C PRDM    Prandtl-Zahl des Dampfes bei (TWK+TDK)*0.5 (-)
C REDM    Reynolds-Zahl des Dampfes bei (TWK+TDK)*0.5 (-)
C REDL    Laminarer Grenzwert der Reynolds-Zahl
C REDT    Turbulenter Grenzwert der Reynolds-Zahl
C TDK     Dampftemperatur (K)
C TWK     Wandtemperatur (K)
C=====
      DOUBLE PRECISION ALPHA, ALPHAL, ALPHAT, DAMUEB, DR, LAMDM, PRDM,
      *      REDM, TDK, TWK, REDL, REDT, DAMLAM, DAMTUR
      LOGICAL LWRITE
      REDL = 2300.
      ALPHAL = DAMLAM (DR, LAMDM, TDK, TWK)
      REDT = 5000.
      ALPHAT = DAMTUR (DR, LAMDM, PRDM, REDT, LWRITE)
      ALPHA = ALPHAL + (REDM - 2300) / 2700 * (ALPHAT - ALPHAL)
      DAMUEB = ALPHA
      IF (LWRITE) THEN
        WRITE (66,*) '-----'
        WRITE (66,*) ' HILFSGROESSEN DAMUEB.F: '
        WRITE (66,999) ' ALPHAL = ', ALPHAL, ' ALPHAT = ', ALPHAT
999    FORMAT (2(A, D15.8, TR6))
        WRITE (66,*) '-----'
      ENDIF
      END

```

## FLULAM

```

      FUNCTION FLULAM (DR, ETAF, ETAFW, LA, LAMF, PRF, REF)
C=====
C Andreas Schaffrath
C Ruhr-Universitaet Bochum / Professur fuer Sicherheitsforschung und
C Reaktortechnik
C Forschungszentrum Juelich
C D-52425 Juelich
C Tel. 02461 / 613437
C 23.1.1995
C=====
C Berechnung des Waermeuebergangskoeffizienten bei laminaren Fluessig-
C keitsstroemung nach Sieder-Tate
C=====
C Variablen:
C ALPHA   Waermeuebergangskoeffizient (W/m2K)
C DR      Durchmesser des Rohres (m)
C ETAF    dynamische Viskositaet der Fluessigkeit (kg/ms)
C ETAFW   dynamische Viskositaet der Fluessigkeit bei der Wandtemperatur
C         (kg/ms)
C FLULAM  WUEK bei laminarer Fluessigkeitsstroemung (W/m2K)
C LA      Laenge des Rohres (m)

```

C LAMF      Waermeleitfaehigkeit der Fluessigkeit (W/mK)  
C PRF      Prandtl-Zahl der Fluessigkeit (-)  
C REF      Reynolds-Zahl der Fluessigkeit (-)

C=====

```
      DOUBLE PRECISION ALPHA, DR, ETAF, ETAFW, LA, LAMF, PRF, REF,
*      FLULAM
      ALPHA = 1.86 * LAMF / DR * (REF * PRF * DR / LA) ** 0.33
*      * (ETAF / ETAFW) ** 0.14
      FLULAM = ALPHA
      END
```

# FLUTUR

```

      FUNCTION FLUTUR (DR, ETAF, ETAFW, LAMF, LAMFM, PRF, PRFM,
*                   REF, REFM, LWRITE)
C=====
C Andreas Schaffrath
C Ruhr-Universitaet Bochum / Professur fuer Sicherheitsforschung und
C Reaktortechnik
C Forschungszentrum Juelich
C D-52425 Juelich
C Tel. 02461 / 613437
C 23.1.1995
C=====
C Berechnung des Waermeuebergangskoeffizienten bei turbulenter
C Fluessigkeitsstroemung nach Dittus-Boelter bzw. Sieder-Tate
C=====
C Variablen:
C ALPHA   Waermeuebergangskoeffizient (W/m2K)
C ALPHAD  Waermeuebergangskoeffizient nach Dittus-Boelter (W/m2K)
C ALPHAS  Waermeuebergangskoeffizient nach Sieder-Tate (W/m2K)
C DR      Durchmesser des Rohres (m)
C ETAF    dynamische Viskositaet der Fluessigkeit (kg/sm)
C ETAFW   dynamische Viskositaet der Fluessigkeit bei der Wandtempera-
C          tur (kg/sm)
C FLUTUR  WUEK bei turbulenter Fluessigkeitsstroemung (W/m2K)
C LAMF    Waermeleitfaehigkeit der Fluessigkeit bei (W/mK)
C LAMFM   Waermeleitfaehigkeit der Fluessigkeit bei (TW+TD)*0.5 (W/mK)
C PRF     Prandtl-Zahl der Fluessigkeit (-)
C PRFM    Prandtl-Zahl der Fluessigkeit bei (TW+TF)*0.5 (-)
C REF     Reynolds-Zahl der Fluessigkeit (-)
C REFM    Reynolds-Zahl der Fluessigkeit bei (TW+TF)*0.5 (-)
C=====
      DOUBLE PRECISION ALPHA, ALPHAD, ALPHAS, DR, ETAF, ETAFW, FLUTUR,
*                   LAMF, LAMFM, PRF, PRFM, REF, REFM
      LOGICAL LWRITE
C-----Dittus-Boelter
      IF (PRFM .LT. 0.7) THEN
        REFM = 0.7
        IF (LWRITE) WRITE (66, '(A)') 'Prandtl-Zahl kleiner 0.7'
      ELSEIF (PRFM .GT. 100.) THEN
        PRFM = 100.
        IF (LWRITE) WRITE (66, '(A)') 'Prandtl-Zahl groesser 100'
      ENDIF
      ALPHAD = 0.023 * LAMFM / DR * REFM ** 0.8 * PRFM ** 0.4
C-----Sieder-Tate
      IF (PRF .LT. 0.6) THEN
        PRF = 0.6
        IF (LWRITE) WRITE (66, '(A)') 'Prandtl-Zahl kleiner 0.6'
      ELSEIF (PRF .GT. 70.) THEN
        PRF = 70.
        IF (LWRITE) WRITE (66, '(A)') 'Prandtl-Zahl groesser 70'
      ENDIF
      ALPHAS = 0.023 * LAMF / DR * REF ** 0.8 * PRF ** 0.33 * (ETAF /
*                   ETAFW) ** 0.14
C-----Ruecksprung
      FLUTUR = MAX (ALPHAD, ALPHAS)
      IF (LWRITE) THEN
        WRITE (66,*) '-----'
        WRITE (66,*) ' HILFSGROESSEN FLUTUR.F:'
        WRITE (66,999) ' ALPHAD = ', ALPHAD, ' ALPHAS = ', ALPHAS
1999  FORMAT (2(A, D15.8, TR6))
        WRITE (66,*) '-----'
      ENDIF
      END

```



## FLUUEB

```

      FUNCTION FLUUEB (DR, ETAF, ETAFW, LA, LAMF, LAMFM, PRF, PRFM,
*                   REF, REFM, LWRITE)
C=====
C Andreas Schaffrath
C Ruhr-Universitaet Bochum / Professur fuer Sicherheitsforschung und
C Reaktortechnik
C Forschungszentrum Juelich
C D-52425 Juelich
C Tel. 02461 / 613437
C 23.1.1995
C=====
C Berechnung des Waermeuebergangskoeffizienten im Uebergangsbereich
C zwischen laminarer und turbulenter Fluessigkeitsstroemung
C=====
C Variablen:
C ALPHA   Waermeuebergangskoeffizient (W/m2K)
C ALPHAD  Waermeuebergangskoeffizient nach Dittus-Boelter (W/m2K)
C ALPHAL  Waermeuebergangskoeffizient laminar (W/m2K)
C ALPHAS  Waermeuebergangskoeffizient nach Sieder-Tate (W/m2K)
C ALPHAT  Waermeuebergangskoeffizient turbulent (W/m2K)
C DR      Durchmesser des Rohres (m)
C ETAF    dynamische Viskositaet der Fluesigkeit (kg/sm)
C ETAFW   dynamsiche Viskositaet der Fluessigkeit bei der Wandtempera-
C          tur (kg/sm)
C FLUUEB  WUEK im Uebergangsbereich bei Fluessigkeitsstroemung (W/m2K)
C LA      Laenge des Rohres (m)
C LAMF    Waermeleitfaehigkeit der Fluessigkeit bei (W/mK)
C LAMFM   Waermeleitfaehigkeit der Fluessigkeit bei (TW+TD)*0.5 (W/mK)
C PRF     Prandtl-Zahl der Fluessigkeit (-)
C PRFM    Prandtl-Zahl der Fluessigkeit bei (TW+TF)*0.5 (-)
C REF     Reynolds-Zahl des Fluessigkeit (-)
C REFM    Reynolds-Zahl des Fluessigkeit bei (TW+TF)*0.5 (-)
C=====
      DOUBLE PRECISION ALPHA, ALPHAD, ALPHAL, ALPHAS, ALPHAT, DR, ETAF,
*                   ETAFW, FLUUEB, LA, LAMF, LAMFM, PRF, PRFM, REF,
*                   REFM, REFL, REFT, FLULAM, FLUTUR
      LOGICAL LWRITE
C-----ALPHA Laminar
      REFL = 2300.
      ALPHAL = FLULAM ( DR, ETAF, ETAFW, LA, LAMF, PRF, REFL)
C-----ALPHA Turbulent
      REFT = 5000.
      ALPHAT = FLUTUR (DR, ETAF, ETAFW, LAMF, LAMFM, PRF, PRFM,
*                   REF, REFT, LWRITE)
C-----Interpolation
      ALPHA = ALPHAL + (REFM - 2300) / 2700 * (ALPHAT - ALPHAL)
      FLUUEB = ALPHA
      IF (LWRITE) THEN
        WRITE (66,*) '-----'
        WRITE (66,*) ' HILFSGROESSEN FLUUEB.F:'
        WRITE (66,999) ' ALPHAL = ', ALPHAL, ' ALPHAT = ', ALPHAT
999  FORMAT (2(A, D15.8, TR6))
        WRITE (66,*) '-----'
      ENDIF
      END

```

## RING

```

      FUNCTION RING (CPF, DHV, DR, ETAFM, ETAFW, ETAFS, HD, HDS, HF,
*           HFS, LAMFM, LAMFS, LAMFW, MGES, NUEFM, P, PRF,
*           PRFM, RODM, ROFM, RED, ROD, ROF, LA, SIGMAM,
*           TSK, TWK, XM, REFM, REF, LWRITE)
C=====
C Andreas Schaffrath
C Ruhr-Universitaet Bochum / Professur fuer Sicherheitsforschung und
C Reaktortechnik
C Forschungszentrum Juelich
C D-52425 Juelich
C Tel. 02461 / 613437
C 30.1.1995
C=====
C Berechnung des Waermeuebergangskoeffizienten der Ringstroemung
C=====
C Variablen:
C ALPHA   Waermeuebergangskoeffizient (W/m2K)
C=====
      DOUBLE PRECISION ALPHA, CPF, DHV, DR, ETAFM,
*           ETAFW, ETAFS, HD, HDS, HF, HFS, LAMFM, LAMFS,
*           LAMFW, MGES, NUEFM, P, PRF, PRFM, RODM, ROFM,
*           RED, ROD, ROF, REFM, REF, RING, LA,
*           SIGMAM, TSK, TWK, XM, REFM, REF
      DOUBLE PRECISION RINLAM, RINTUR, RINUEB
      LOGICAL          LWRITE
      IF (LWRITE) THEN
        WRITE (66,*) '-----'
        WRITE (66,*) ' HILFSGROESSEN RINUEB.F: '
      ENDIF

C
C   Verzweigung zu laminarer oder turbulenter Ringstroemung
C
      IF (REFM .LE. 100./PRFM) THEN
C
C   Laminare Ringstroemung
C
        IF (LWRITE) THEN
          WRITE (66,*) 'Laminare Ringstroemung'
          WRITE (66,999) ' REFM = ', REFM, ' 100./PRFM = ',
*                      100./PRFM
        ENDIF
        ALPHA = RINLAM (DHV, ETAFM, ETAFW, ETAFS, HD, HDS, HF, HFS,
*                      LA, LAMFM, LAMFS, LAMFW, REFM, RODM, ROFM,
*                      SIGMAM, TSK, TWK, NUEFM, LWRITE )
      ELSEIF (REFM .GE. 500./PRFM) THEN
C
C   Turbulente Ringstroemung
C
        IF (LWRITE) THEN
          WRITE (66,*) 'Turbulente Ringstroemung'
          WRITE (66,999) ' REFM = ', REFM, ' 500./PRFM = ',
*                      500./PRFM
        ENDIF
        ALPHA = RINTUR (CPF, DR, MGES, PRF, RED, REF, ROD,
*                      ROF, XM, P, LWRITE)
      ELSE
C
C   Uebergang zwischen laminarer und turbulenter Ringstroemung
C
        IF (LWRITE) WRITE (66,*) 'Ringstroemung Uebergang lam - turb'
        ALPHA = RINUEB (CPF, DHV, DR, ETAFM, ETAFW, ETAFS, HD,
*                      HDS, HF, HFS, LAMFM, LAMFS, LAMFW,
*                      MGES, NUEFM, P, PRF, PRFM, REFM,
*                      RODM, ROFM, RED, ROD, ROF, LA,
*                      SIGMAM, TSK, TWK, XM, LWRITE)
      ENDIF
      RING = ALPHA

```

```
999  FORMAT (2(A, D15.8, TR6))  
      IF (LWRITE) WRITE (66,*)
```

```
*      '-----'  
      END
```

# RINLAM

```

      FUNCTION RINLAM (DHV, ETAFM, ETAFW, ETAFS, HD, HDS, HF, HFS,
*                LAENGE, LAMFM, LAMFS, LAMFW, REFM, RODM, ROFM,
*                SIGMAM, TSK, TWK, NUEFM, LWRITE )
C=====
C Andreas Schaffrath
C Ruhr-Universitaet Bochum / Professur fuer Sicherheitsforschung und
C Reaktortechnik
C Forschungszentrum Juelich
C D-52425 Juelich
C Tel. 02461 / 613437
C 13.1.1995
C=====
C Berechnung des Waermeuebergangskoeffizienten bei laminarer Ring-
C stroemungstroemung nach Nusselt
C=====
C Variablen:
C ALPHA    Waermeuebergangskoeffizient (W/m2K)
C ALPHAD   Waermeuebergangskoeffizient nach Dittus-Boelter (W/m2K)
C ALPHAS   Waermeuebergangskoeffizient nach Sieder-Tate (W/m2K)
C DR       Durchmesser des Rohres (m)
C ETAF     dynamische Viskositaet der Fluessigkeit (kg/sm)
C ETAFW    dynamische Viskositaet der Fluessigkeit bei der Wandtempera-
C           tur (kg/sm)
C LAMF     Waermeleitfaehigkeit der Fluessigkeit bei (W/mK)
C LAMFM    Waermeleitfaehigkeit der Fluessigkeit bei (TWK+TDK)*0.5 (W/mK)
C PRF      Prandtl-Zahl der Fluessigkeit (-)
C PRFM     Prandtl-Zahl der Fluessigkeit bei (TWK+TFK)*0.5 (-)
C REF      Reynolds-Zahl des Fluessigkeit (-)
C REFM     Reynolds-Zahl des Fluessigkeit bei (TWK+TFK)*0.5 (-)
C ALPHANU  Waermeuebergangskoeffizient nach Nusselt (W/m2K)
C DH       Enthlpiedifferenz zwischen Dampf und Kondensat (J/kg)
C DHU      Enthapliedifferenz der Fluessigkeit zwischen HFS und HF (J/kg)
C DHUE     Enthapliedifferenz des Dampfes zwischen HD und HDS (J/kg)
C DHV      Verdampfungsenthalpie (J/kg)
C DR       Durchmesser des Rohres (m)
C EPS      vol. Dampfgehalt (-)
C ETAFS    dynamische Viskositaet der Fluessigkeit bei der Saetti-
C           gung (kg/sm)
C ETAST     normierte dynamische Viskositaet der Fluessigkeit (-)
C ETAFW    dynamische Viskositaet der Fluessigkeit bei der Wandtempera-
C           tur (kg/sm)
C FSCHICH  Korrekturfaktor zur Beruecksichtigung des Einflusses des
C           Sumpfes auf den WUEK (-)
C FT       Korrekturfaktor zur Beruecksichtigung der Temperaturabhaen-
C           gigkeit der Stoffwerte des Films (-)
C HD       Enthalpie des Dampfes (J/kg)
C HDS      Enthalpie des Dampfes bei Saettigung (J/kg)
C HF       Enthalpie der Fluessgkeit (J/kg)
C HFS      Enthalpie der Fluessgkeit bei Saettigungstemperatur (J/kg)
C LAMF     Waermeleitfaehigkeit der Fluessigkeit bei (W/mK)
C LAMFM    Waermeleitfaehigkeit der Fluessigkeit bei (TWK+TDK)*0.5 (W/mK)
C LAMST    normierte Waermeleitfaehigkeit der Fluessigkeit (-)
C NUEFM    kinematische Viskositaet bei (TWK+TFK)*0.5 (m2/s)
C PRF      Prandtl-Zahl der Fluessigkeit (-)
C PRFM     Prandtl-Zahl der Fluessigkeit bei (TWK+TFK)*0.5 (-)
C REF      Reynolds-Zahl des Fluessigkeit (-)
C REFM     Reynolds-Zahl des Fluessigkeit bei (TWK+TFK)*0.5 (-)
C RODM     Dichte des Dampfes bei (TWK+TDK)*0.5 (kg/m3)
C ROFM     Dichte des Kondensatfilms bei (TWK+TFK)*0.5 (kg/m3)
C SCHLAM   WUEK bei Schichtenstroemung (W/m2K)
C TSK      Saettigungstemperatur (K)
C TWK      Wandtemperatur (K)
C=====
      DOUBLE PRECISION RINLAM, ALPHA, ALPHAN1, ALPHAN2
      DOUBLE PRECISION DHU, HFS, HF, DHUE, HD, HDS, DH, DHV
      DOUBLE PRECISION ROFM, RODM, LAMFM, ETAFM
      DOUBLE PRECISION LAENGE, TSK, TWK, REFM, ALPHANU

```

```

DOUBLE PRECISION REWELL, SIGMAM, NUEFM, FWELL
DOUBLE PRECISION FT, ETAST, ETAFS, ETAFW, LAMST, LAMFS, LAMFW
LOGICAL LWRITE
C-----Nusselt
C-----Dampfueberhitzung bzw. Kondensatunterkuehlung
DHU = 0.68 * (HFS - HF)
DHUE = HD - HDS
DH = DHV + DHUE + DHU
C-----WUEK in Abhaengigkeit von der Laenge des Rohres
ALPHAN1 = ((ROFM * (ROFM - RODM) * 9.81 * DH * LAMFM ** 3) /
* (4. * ETAFM * LAENGE * (TSK - TWK))) ** 0.25
C-----Nusselt in Abhaengigkeit vom Kondensatfilm
ALPHAN2 = (1.1025 * LAMFM * 1.0 / (REFM ** 0.33)) *
* ((ROFM * (ROFM - RODM) * 9.81) / ETAFM ** 2) ** 0.33
ALPHANU = MAX (ALPHAN1, ALPHAN2)
C-----Wellen
C-----Wellenreynoldszahl nach Grimley
REWELL = 0.392 * ( (SIGMAM / (ROFM * 9.81)) ** 0.5 * (9.81 /
* NUEFM ** 2. ) ** 0.33 ) ** 0.75
C-----Korrekturfaktor fwell nach Zazuli
FWELL = 1.0
IF (REFM .GT. REWELL) THEN
    FWELL = 0.8 * REFM ** 0.11
ENDIF
C-----Temperaturabhaengige Stoffwerte
FT = 1.0
ETAST = ETAFS / ETAFW
LAMST = LAMFS / LAMFW
FT = (1. + ETAST) / (10. * (1. + LAMST) ** 3.) * (5. + LAMST * (
* 14. + 11. * LAMST) + LAMST / ETAST * (1. + 4. * LAMST +
* 5. * LAMST ** 2.))
C-----ALPHA
ALPHA = ALPHANU * FWELL * FT
RINLAM = ALPHA
C-----HILFSAUSGABEN
IF (LWRITE) THEN
WRITE (66,*) '-----'
WRITE (66,*) ' HILFSGROESSEN RINLAM.F: '
WRITE (66,999) ' DHU      = ', DHU, ' DHUE      = ', DHUE
WRITE (66,999) ' DH      = ', DH, ' ALPHANU = ', ALPHANU
WRITE (66,999) ' ALPHAN1 = ', ALPHAN1, ' ALPHAN2 = ', ALPHAN2
WRITE (66,999) ' REWELL  = ', REWELL, ' FWELL   = ', FWELL
WRITE (66,999) ' ETAST   = ', ETAST, ' LAMST   = ', LAMST
WRITE (66, '(A, D15.8)') ' FT      = ', FT
999 FORMAT (2(A, D15.8, TR6))
WRITE (66,*) '-----'
ENDIF
END

```

## RINTUR

```

      FUNCTION RINTUR (CPF, DR, MGES, PRF, RED, REF, ROD, ROF, XM, P
      *
      , LWRITE)
C=====
C Andreas Schaffrath
C Ruhr-Universitaet Bochum / Professur fuer Sicherheitsforschung und
C Reaktortechnik
C Forschungszentrum Juelich
C D-52425 Juelich
C Tel. 02461 / 613437
C 13.1.1995
C=====
C Berechnung des Waermeuebergangskoeffizienten bei turbulenter
C Ringstroemung nach Kosky und Staub
C=====
C Variablen:
C ALPHA   Waermeuebergangskoeffizient (W/m2K)
C CD      Konstante fuer Druckverlustbestimmung GE 402.4 S. 14
C CF      Konstante fuer Druckverlustbestimmung GE 402.4 S. 14
C CPF     spez. Waermekapazitaet der Fluessigkeit (J/kgK)
C DPF     Druckverlust der einphasigen Fluessigkeitsstroemung (Pa/m)
C DPLUS   dimensionslose Filmdicke (-)
C DPTP    Druckverlust der Zweiphasenstroemung (Pa/m)
C DR      Durchmesser des Rohres (m)
C M       Exponent in Brebber Korrelation ( M = 0.45)
C MGES    Massenstrom gesamt (Dampf & Fluessigkeit) (kg/s)
C N       Konstante fuer Druckverlustbestimmung GE 402.4 S. 14
C ND      Konstante fuer Druckverlustbestimmung GE 402.4 S. 14
C NF      Konstante fuer Druckverlustbestimmung GE 402.4 S. 14
C PI      Konstante (=3.14159)
C PHI2    Zweiphasenmultiplikator
C PRF     Prandtl-Zahl der Fluessigkeit (-)
C RED     Reynolds-Zahl des Dampfes (-)
C REF     Reynolds-Zahl der Fluessigkeit (-)
C RINTUR  WUEK bei zurbulenter Rinststroemung (W/m2K)
C ROD     Dichte des Dampfes (kg/m3)
C ROF     Dichte der Fluessigkeit (kg/m3)
C TPLUS   dimensionslose Temperatur (-)
C VSTERN  Dampfgeschwindigkeit an der Phasengrenze (m/s)
C WF      Kondensatfilmgeschwindigkeit unter der Annahme. dass das
C          Kondensat allein im Rohrquerschnitt stroemt
C X       Wurzel Martinelli-Nelson Parameter
C X2      Martinelli-Nelson Parameter
C XM      Dampfmassengehalt
C=====
      DOUBLE PRECISION ALPHA, CD, CF, DR, M, MGES, N, ND, NF, PI,
      *
      PHI2, PRF, RED, REF, RINTUR, ROD, ROF, TPLUS,
      *
      VSTERN, WF, X, X2, XM, P, CPF
      LOGICAL          LWRITE
      DATA M /0.45/
      DATA PI /3.14159/
C-----Bestimmung der dimensionalsosen Kondensatfilmdicke
C-----Fallunterscheidung REF laminar oder turbulent
      DPLUS = 0.0
      IF (REF .GE. 1000.) THEN
C----- turbulenter Kondensatfilm
          DPLUS = 0.0504 * REF**(7./8.)
      ELSE
C----- laminarer Kondensatfilm
          DPLUS = 0.0504 *( 0.5* REF) ** 0.5
      ENDIF
C-----Bestimmung des dim. Temperaturprofiles in Abhaengigkeit
C-----von der dimensionslosen Kondensatfilmdicke
      IF (DPLUS .LE. 5.0) THEN
          TPLUS = PRF * DPLUS
      ELSEIF (DPLUS .GE. 30.0) THEN
          TPLUS = 5. * (PRF + LOG (1. + 5. * PRF) + 0.495 * LOG (DPLUS
      *
      / 30.))

```

```

ELSE
  TPLUS = 5. * (PRF + LOG (1 + PRF * (DPLUS / 5. - 1.)))
ENDIF
C-----Bestimmung der Dampfgeschwindigkeit an der Phasengrenze
C-----Bestimmung der Kondensatgeschwindigkeit unter der Annahme,
C-----dass alles Kondensat allein im Rohrquerschnitt stroemt
  WF = 4.*(1. - XM) * MGES / ROF / PI / DR ** 2.
C-----Bestimmung des einphasigen Druckverlustes unter der Annahme,
C-----dass alles Kondensat allein im Rohrquerschnitt stroemt
  DPF = 0.02 * ROF / DR * WF * WF / 2.
C-----Bestimmung des Zweiphasen-Parameters PHIF2
  IF (RED .GE. 2000.) THEN
    ND = 0.25
    CD = 0.079
  ELSE
    ND = 1.0
    CD = 16.0
  ENDIF
  IF (REF .GE. 2000.) THEN
    NF = 0.25
    CF = 0.079
  ELSE
    NF = 1.0
    CF = 16.0
  ENDIF
  IF (REF .GE. 1000. .AND. RED .GE. 1000.) THEN
    IF (P .LE. 1.1D+06) THEN
      N = 5.13
    ELSE
      N = 4.0
    ENDIF
  ELSEIF (REF .LT. 1000. .AND. RED .LT. 1000.) THEN
    N = 2.75
  ELSE
    N = 3.5
  ENDIF
  X2 = (RED ** ND / REF ** NF) * CF / CD * ROD / ROF *
  * ((1. - XM) / XM) ** 2.
  X = sqrt (X2)
  PHIF2 = (1. + (1. / X) ** (2. / N)) ** N
C-----Bestimmung des Druckverlustes der Zweiphasenstroemung
  DPTP = PHIF2 * DPF
C-----Dampfgeschwindigkeit an der Phasengrenze
  VSTERN = SQRT (DR / 4. / ROF * DPTP)
C-----WUEK nach Kosky und Staub
  ALPHA = ROF * VSTERN * CPF / TPLUS
  RINTUR = ALPHA
  IF (LWRITE) THEN
    WRITE (66,*) '-----'
    WRITE (66,*) ' HILFSGROESSEN RINTUR.F:'
    WRITE (66,999) ' DPLUS = ', DPLUS, ' TPLUS = ', TPLUS
    WRITE (66,999) ' WF = ', WF, ' DPF = ', DPF
    WRITE (66,999) ' N = ', N, ' M = ', M
    WRITE (66,999) ' ND = ', ND, ' NF = ', NF
    WRITE (66,999) ' CF = ', CF, ' CD = ', CD
    WRITE (66,999) ' X2 = ', X2, ' X = ', X
    WRITE (66,999) ' PHIF2 = ', PHIF2, ' DPTP = ', DPTP
    WRITE (66, '(A, D15.8)') ' VSTERN = ', VSTERN
999 FORMAT (2(A, D15.8, TR6))
    WRITE (66,*) '-----'
  ENDIF
END

```

# RINUEB

```

      FUNCTION RINUEB (CPF, DHV, DR, ETAFM, ETAFW, ETAFS, HD, HDS, HF,
*                   HFS, LAMFM, LAMFS, LAMFW, MGES, NUEFM, P, PRF,
*                   PRFM, REFM, RODM, ROFM, RED, ROD, ROF, LA,
*                   SIGMAM, TSK, TWK, XM, LWRITE)
C=====
C Andreas Schaffrath
C Ruhr-Universitaet Bochum / Professur fuer Sicherheitsforschung und
C Reaktortechnik
C Forschungszentrum Juelich
C D-52425 Juelich
C Tel. 02461 / 613437
C 30.1.1995
C=====
C Berechnung des Waermeuebergangskoeffizienten im Uebergangsbereich
C zwischen laminarer und turbulenter Ringstroemung
C=====
C Variablen:
C ALPHA   Waermeuebergangskoeffizient (W/m2K)
C ALPHAL  Waermeuebergangskoeffizient laminar (W/m2K)
C ALPHAT  Waermeuebergangskoeffizient turbulent (W/m2K)
C=====
      DOUBLE PRECISION ALPHAL, ALPHAT, ALPHA, CPF, DHV, DR, ETAFM,
*                   ETAFW, ETAFS, HD, HDS, HF, HFS, LAMFM, LAMFS,
*                   LAMFW, MGES, NUEFM, P, PRF, PRFM, RODM, ROFM,
*                   RED, ROD, ROF, RINUEB, REFIL, REFIT, LA,
*                   SIGMAM, TSK, TWK, XM, REFM
      DOUBLE PRECISION RINLAM, RINTUR
      LOGICAL        LWRITE
C-----ALPHA Laminar
      REFIL = 100./PRFM
      ALPHAL = RINLAM (DHV, ETAFM, ETAFW, ETAFS, HD, HDS, HF, HFS,
*                   LA, LAMFM, LAMFS, LAMFW, REFIL, RODM, ROFM,
*                   SIGMAM, TSK, TWK, NUEFM, LWRITE )
C-----ALPHA Turbulent
      REFIT = 500./PRFM
      ALPHAT = RINTUR (CPF, DR, MGES, PRF, RED, REFIT, ROD,
*                   ROF, XM, P, LWRITE)
C-----Interpolation
      ALPHA = ALPHAL+(REFM-REFIL)/(REFIT-REFIL)*(ALPHAT - ALPHAL)
      RINUEB = ALPHA
      IF (LWRITE) THEN
        WRITE (66,*) '-----'
        WRITE (66,*) ' HILFSGROESSEN RINUEB.F: '
        WRITE (66,999) ' REFM = ', REFM
        WRITE (66,999) ' 100./PRFM = ', REFIL, ' 500./PRFM = ', REFIT
        WRITE (66,999) ' ALPHAL = ', ALPHAL, ' ALPHAT = ', ALPHAT
999  FORMAT (2(A, D15.8, TR6))
        WRITE (66,*) '-----'
      ENDIF
      END

```



# SPRUEH

```

      FUNCTION SPRUEH (CPD, DHV, DR, ETAD, ETAF, GG, LAMF, PRD,
*                TSK, TWK, XM, LWRITE)
C=====
C Andreas Schaffrath
C Ruhr-Universitaet Bochum / Professur fuer Sicherheitsforschung und
C Reaktortechnik
C Forschungszentrum Juelich
C D-52425 Juelich
C Tel. 02461 / 613437
C 23.1.1995
C=====
C Berechnung des Waermeuebergangskoeffizienten bei Spruehstroemung
C nach Solimann
C=====
C Variablen:
C ALPHA   Waermeuebergangskoeffizient (W/m2K)
C CPD      spezifische Waermekapazitaet des Dampfes (J/kgK)
C DHV      Verdampfungsenthalpie (J/kgK)
C DR       Rohrdurchmesser (m)
C ETAD     dynamische Viskositaet des Dampfes (kg/sm)
C ETAF     dynamische Viskositaet der Fluessigkeit (kg/sm)
C ETAM     dynamsiche Viskositaet des Gemisches (kg/sm)
C GG       Gesamtmassenstromdichte (kg/m2s)
C LAMF     Waermeleitfaehigkeit der Fluessigkeit bei (W/mK)
C PRD      Prandtl-Zahl des Dampfes (-)
C REM      Reynolds-Zahl des Gemisches (-)
C TSK      Saettigungstemperatur (K)
C TWK      Wandtemperatur (K)
C XM       Massenstrom-Dampfgehalt (-)
C=====
      DOUBLE PRECISION ALPHA, CPD, DHV, DR, ETAD, ETAF, ETAM, GG, LAMF,
*                PRD, REM, TSK, TWK, XM, SPRUEH
      LOGICAL
      LWRITE
C-----Berechnung der dynamischen Viskositaet des Gemisches
      ETAM = ((XM / ETAD) + (1. - XM) / ETAF)
      ETAM = 1./ETAM
C-----Berechnung der Reynolds-Zahl des Gemisches
      REM = GG * DR / ETAM
C-----Berechnung des Waermeuebergangskoeffizienten
      ALPHA = 0.00345 * LAMF / DR * REM**0.9 * PRD**0.33 * 1./(((TSK -
*                TWK) * CPD / DHV)**0.33)
      SPRUEH = ALPHA
      IF (LWRITE) THEN
        WRITE (66,*) '-----'
        WRITE (66,*) ' HILFSGROESSEN SPRUEH.F: '
        WRITE (66,999) ' ETAM = ', ETAM, ' REM = ', REM
999  FORMAT (2(A, D15.8, TR6))
        WRITE (66,*) '-----'
      ENDIF
      END

```

# SCHLAM

```

      FUNCTION SCHLAM (DHV, DR, EPS, ETAFM, ETAFS, ETAFW, HD, HDS, HF,
*                   HFS, LAMFM, LAMFS, LAMFW, ROFM, TSK, TWK,
*                   LWRITE)
C=====
C Andreas Schaffrath
C Ruhr-Universitaet Bochum / Professur fuer Sicherheitsforschung und
C Reaktortechnik
C Forschungszentrum Juelich
C D-52425 Juelich
C Tel. 02461 / 613437
C 23.1.1995
C=====
C Berechnung des Waermeuebergangskoeffizienten bei Schichtenstroemung
C nach Jaster und Kosky bzw. Rufer und Kezios
C=====
C Variablen:
C ALPHANU Waermeuebergangskoeffizient nach Nusselt (W/m2K)
C DH      Enthalpiedifferenz zwischen Dampf und Kondensat (J/kg)
C DHU     Enthalpiedifferenz der Fluessigkeit zwischen HFS und HF (J/kg)
C DHUE    Enthalpiedifferenz des Dampfes zwischen HD und HDS (J/kg)
C DHV     Verdampfungsenthalpie (J/kg)
C DR      Durchmesser des Rohres (m)
C EPS     vol. Dampfgehalt (-)
C ETAFM   dynamische Viskositaet der Fluessigkeit bei (TWK+TFK)*0.5 (kg/sm)
C ETAFS   dynamische Viskositaet der Fluessigkeit bei der Saetti-
C         gung (kg/sm)
C ETAST   normierte dynamische Viskositaet der Fluessigkeit (-)
C ETAFW   dynamische Viskositaet der Fluessigkeit bei der Wandtempera-
C         tur (kg/sm)
C FSCHICH Korrekturfaktor zur Beruecksichtigung des Einflusses des
C         Sumpfes auf den WUEK (-)
C FT      Korrekturfaktor zur Beruecksichtigung der Temperaturabhaen-
C         gigkeit der Stoffwerte des Films (-)
C HD      Enthalpie des Dampfes (J/kg)
C HDS     Enthalpie des Dampfes bei Saettigung (J/kg)
C HF      Enthalpie der Fluessigkeit (J/kg)
C HFS     Enthalpie der Fluessigkeit bei Saettigungstemperatur (J/kg)
C LAMFM   Waermeleitfaehigkeit der Fluessigkeit bei (TWK+TDK)*0.5 (W/mK)
C LAMFS   Waermeleitfaehigkeit der Fluessigkeit bei Tsaettigung (W/mK)
C LAMFW   Waermeleitfaehigkeit der Fluessigkeit bei TWand (W/mK)
C LAMST   normierte Waermeleitfaehigkeit der Fluessigkeit (-)
C ROFM    Dichte des Kondensatfilms bei (TWK+TFK)*0.5 (kg/m3)
C SCHLAM  WUEK bei Schichtenstroemung (W/m2K)
C TSK     Saettigungstemperatur (K)
C TWK     Wandtemperatur (K)
C=====
      DOUBLE PRECISION ALPHANU, DH, DHU, DHUE, DHV, DR, EPS, ETAFM,
*                   ETAFS, ETAST, ETAFW, FSCHICH, FT, HD, HDS, HF,
*                   HFS, LAMFM, LAMFS, LAMFW, LAMST, ROFM,
*                   SCHLAM, TSK, TWK
      LOGICAL LWRITE
C-----Berechnung des ALPHA Wertes nach Nusselt
      DHU = 0.68 * (HFS - HF)
      DHUE = HD - HDS
      IF (LWRITE) WRITE (66,999)
*      ' DHU = ', DHU, ' DHUE = ', DHUE
      DH = DHV + DHUE + DHU
      ALPHANU= 0.7251 * ((ROFM**2. * 9.81 * DH * LAMFM**3.) / (ETAFM*
*      DR * (TSK - TWK))**0.25
      IF (LWRITE) WRITE (66,999) ' DH = ', DH,
*      ' ALPHANU = ', ALPHANU
C-----Berechnung des Korrekturfaktors fuer temperaturabhaen. Stoffwerte
      FT = 1.0
      ETAST = ETAFS / ETAFW
      LAMST = LAMFS / LAMFW
      IF (LWRITE) WRITE (66,999) ' ETAST = ', ETAST,
*      ' LAMST = ', LAMST

```

```

      FT = (1. + ETAST) / (10. * ( 1. + LAMST)**3.) * (5. + LAMST *
*      (14. + 11. * LAMST) + LAMST / ETAST * (1. + 4.* LAMST +
*      5. * LAMST**2.))
C-----Berechnung des lokalen Korrekturfaktors fuer Schichtenstroemung
      FSCHICH = EPS ** 0.75
      IF (LWRITE) WRITE (66,999) ' FT          = ', FT,
*                                ' FSCHICH = ', FSCHICH
C-----Waermeuebergangskoeffizient bei laminarer Schichtenstroemung
      SCHLAM = FSCHICH * FT * ALPHANU
      IF (LWRITE) THEN
        WRITE (66,*) '-----'
        WRITE (66,*) ' HILFSGROESSEN SCHLAM.F:'
        WRITE (66,999) ' DHU          = ', DHU, ' DHUE          = ', DHUE
        WRITE (66,999) ' DH           = ', DH, ' ALPHANU = ', ALPHANU
        WRITE (66,999) ' ETAST        = ', ETAST, ' LAMST        = ', LAMST
        WRITE (66,999) ' FT           = ', FT, ' FSCHICH = ', FSCHICH
999  FORMAT (2(A, D15.8, TR6))
        WRITE (66,*) '-----'
      ENDIF
      END

```



## **Anhang C: Listing von PROP**

```

      SUBROUTINE PROP(P, TD, TF, TW, TS, TDW, TFW, CPD, CPDM, CPF, CPFM,
      *              DHV, ETAD, ETADM, ETAF, ETAFM, ETAFS, ETAFW, HD,
      *              HDS, HF, HFS, LAMD, LAMDM, LAMF, LAMFM, LAMFS,
      *              LAMFW, NUEFM, PRD, PRDM, PRF, PRFM, SIGMAM, ROD,
      *              RODM, ROF, ROFM, LWRITE)
C=====
C Andreas Schaffrath
C Ruhr-Universitaet Bochum / Professur fuer Sicherheitsforschung und
C Reaktortechnik
C Forschungszentrum Juelich
C D-52425 Juelich
C Tel. 02461 / 613437
C 23.1.1995
C=====
CZ-----
CZ  Zweck:  Stoffwerteberechnung fuer KONWAR
CZ-----
C
C  ACHTUNG
C  DRUCKEINGABE IN PA!!
C  FUNKTIONSAUFRUF MPATS ERGIBT TEMPERATUR IN K!!!
C  FUNKTIONSAUFRUF MPTAF ERFORDERT TEMPERATUR IN C!!!
C
CF  Eingabeparameter:
CF
CF  P      Druck in (Pa)
CF  TD     Dampftemperatur in (C)
CF  TF     Fluessigkeitstemperatur in (C)
CF  TW     Wandtemperatur in (C)
CF
CF  Ausgabeparameter:
CF
CF  TS     Saettigungstemperatur in (K) !!!!!
CF  TDW    Mittelwert aus Dampf- und Wandtemperatur in (C)
CF  TFW    Mittelwert aus Fluessigkeits- und Wandtemperatur in (C)
CF  CPD    Spez. Waermekapazitaet des Dampfes bei TD in (J/kgK)
CF  CPDM   Spez. Waermekapazitaet des Dampfes bei TS in (J/kgK)
CF  CPF    Spez. Waermekapazitaet der Fluessigkeit bei TF in (J/kgK)
CF  CPFM   Spez. Waermekapazitaet der Fluessigkeit bei TFW in (J/kgK)
CF  DHV    Verdampfungsenthalpie in (J/kg)
CF  ETAD   dynamische Viskositaet des Dampfes bei TD in (kg/ms)
CF  ETADM  dynamische Viskositaet des Dampfes bei TS in (kg/ms)
CF  ETAF   dynamische Viskositaet der Fluessigkeit bei TF in (kg/ms)
CF  ETAFM  dynamische Viskositaet der Fluessigkeit bei TFW in (kg/ms)
CF  ETAFS  dynamische Viskositaet der Fluessigkeit bei TS in (kg/ms)
CF  ETAFW  dynamische Viskositaet der Fluessigkeit bei TW in (kg/ms)
CF  HD     Spez. Enthalpie des Dampfes bei TD in (J/kg)
CF  HDS    Spez. Enthalpie des Dampfes bei TS in (J/kg)
CF  HF     Spez. Enthalpie der Fluessigkeit bei TF in (J/kg)
CF  HFS    Spez. Enthalpie der Fluessigkeit bei TS in (J/kg)
CF  LAMD   Waermeleitfaehigkeit des Dampfes bei TD in (W/mK)
CF  LAMDM  Waermeleitfaehigkeit des Dampfes bei TS in (W/mK)
CF  LAMF   Waermeleitfaehigkeit der Fluessigkeit bei TF in (W/mK)
CF  LAMFM  Waermeleitfaehigkeit der Fluessigkeit bei TFW in (W/mK)
CF  LAMFS  Waermeleitfaehigkeit der Fluessigkeit bei TS in (W/mK)
CF  LAMFW  Waermeleitfaehigkeit der Fluessigkeit bei TW in (W/mK)
CF  NUEFM  kinematische Viskositaet der Fluessigkeit bei TFW in (m2/s)
CF  PRD    Prandtlzahl des Dampfes bei TD
CF  PRDM   Prandtlzahl des Dampfes bei TS
CF  PRF    Prandtlzahl der Fluessigkeit bei TF
CF  PRFM   Prandtlzahl der Fluessigkeit bei TFW
CF  SIGMAM Oberflaechenspannung der Fluessigkeit bei TFW in (N/m)
CF  ROD    Dichte des Dampfes bei TD in (kg/m3)
CF  RODM   Dichte des Dampfes bei TS in (kg/m3)
CF  ROF    Dichte der Fluessigkeit bei TF in (kg/m3)
CF  ROFM   Dichte der Fluessigkeit bei TFW in (kg/m3)
C-----
C      INCLUDE 'c'
C      INCLUDE 'ca'
C-----
C  Vereinbarung der formalen Parameter
C-----

```

```

      DOUBLE PRECISION P, TD, TF, TW
      DOUBLE PRECISION TSK, TS, TDW, TFW, CPD, CPDM, CPF, CPFM, DHV,
*          ETAD, ETADM, ETAF, ETAFM, ETAFS, ETAFW,
*          HD, HDS, HF, HFS, LAMD, LAMDM, LAMF, LAMFM,
*          LAMFS, LAMFW, NUEFM, PRD, PRDM, PRF, PRFM,
*          SIGMAM, ROD, RODM, ROF, ROFM
      LOGICAL          LWRITE
C-----
CF  Verwendete Unterprogramme
CF
CF  direkter Aufruf von:
CF  MPATS, MPTAF, MPAHFG
CF
CF  indirekte Aufrufe von:
CF  MPTSA, S, SNDS, SPOLY
C-----
      DOUBLE PRECISION MPATS, MPTAF, MPAHFG, MPTSA, S, SPOLY, SNDS
C-----
C-----Ausfuehrungsteil-----
C-----
C-----Saettigungstemperatur
      TSK = MPATS (P)
      TS  = TSK - 273.15
C-----
C-----Stoffwerte fuer die Fluessigkeit
C-----
C-----bei der Temperatur T = TS
      ETAFS = MPTAF (107, P, TS)
      HFS   = 1000. * MPTAF (102, P, TS)
      DHV   = MPAHFG (P)
C-----bei der Temperatur T = 0.5 * (TW + TF)
      TFW = 0.5 * (TW + TF)
      LAMFM = 1000. * MPTAF (106, P, TFW)
      ROFM  = 1./MPTAF (101, P, TFW)
      SIGMAM = MPTAF (108, P, TFW)
      ETAFM  = MPTAF (107, P, TFW)
      CPFM   = 1000. * MPTAF (105, P, TFW)
      PRFM   = ETAFM * CPFM / LAMFM
      NUEFM  = ETAFM / ROFM
C-----bei der Temperatur T = TW
      ETAFW = MPTAF (107, P, TW)
      LAMFW = 1000. * MPTAF (106, P, TW)
C-----bei der Temperatur T = TF
      LAMF  = 1000. * MPTAF (106, P, TF)
      ROF   = 1./MPTAF (101, P, TF)
      ETAF  = MPTAF (107, P, TF)
      CPF   = 1000. * MPTAF (105, P, TF)
      PRF   = ETAF * CPF / LAMF
      HF    = 1000. * MPTAF (102, P, TF)
C-----
C-----Stoffwerte fuer den Dampf
C-----
C-----bei der Temperatur T = TS
      HDS   = MPTAF (-102, P, TS)
      TDW   = 0.5 * (TW + TD)
      IF (TD .LT. TS) TD = TS
      IF (TDW .LT. TS) TDW = TS
C-----bei der Temperatur TDM = TD
      LAMDM = 1000. * MPTAF (-106, P, TDW)
      RODM  = 1./MPTAF (-101, P, TDW)
      ETADM = MPTAF (-107, P, TDW)
      CPDM  = 1000. * MPTAF (-105, P, TDW)
      PRDM  = ETADM * CPDM / LAMDM
C-----bei der Temperatur T = TD
      LAMD  = 1000. * MPTAF (-106, P, TD)
      ROD   = 1./MPTAF (-101, P, TD)
      ETAD  = MPTAF (-107, P, TD)
      CPD   = 1000. * MPTAF (-105, P, TD)
      PRD   = ETAD * CPD / LAMD
      HD    = 1000. * MPTAF (-102, P, TD)
C-----
C----- AUSGABE

```

```

C-----
      IF (LWRITE) THEN
      WRITE (66, *) '++++++'
      WRITE (66, *) '++++++'
      WRITE (66, *) '          STOFFWERTEBERECHNUNG'
      WRITE (66, *) '++++++'
      WRITE (66, *) '          EINGABEGROESSEN:'
      WRITE (66, *) '++++++'
      WRITE (66, 9) ' P      = ', P, ' in Pa'
      WRITE (66, 9) ' TD      = ', TD, ' in C'
      WRITE (66, 9) ' TW      = ', TW, ' in C'
      WRITE (66, 9) ' TF      = ', TF, ' in C'
      WRITE (66, *) '++++++'
      WRITE (66, *) '          AUSGABEGROESSEN:'
      WRITE (66, *) '++++++'
      WRITE (66, 9) ' TS      = ', TSK, ' in K'
      WRITE (66, 9) ' TS      = ', TS, ' in C'
      WRITE (66, 9) ' TDW     = ', TDW, ' in C'
      WRITE (66, 9) ' TFW     = ', TFW, ' in C'
      WRITE (66, *) '*****'
      WRITE (66, *) '          STOFFDATEN BEI SAETTIGUNGSTEMPERATUR TS:'
      WRITE (66, *) '*****'
      WRITE (66, 9) ' ETAFS   = ', ETAFS, ' in kg/ms'
      WRITE (66, 9) ' HFS     = ', HFS, ' in J/kg'
      WRITE (66, 9) ' HDS     = ', HDS, ' in J/kg'
      WRITE (66, 9) ' DHV     = ', DHV, ' in J/kg'
      WRITE (66, *) '*****'
      WRITE (66, *) '          STOFFDATEN BEI 0.5 * (TW + TF):'
      WRITE (66, *) '*****'
      WRITE (66, 9) ' LAMFM   = ', LAMFM, ' in W/mK'
      WRITE (66, 9) ' ROFM    = ', ROFM, ' in kg/m3'
      WRITE (66, 9) ' SIGMAM  = ', SIGMAM, ' in N/m'
      WRITE (66, 9) ' ETAFM   = ', ETAFM, ' in kg/ms'
      WRITE (66, 9) ' CPFM    = ', CPFM, ' in J/kgK'
      WRITE (66, 9) ' PRFM    = ', PRFM, ' in (-)'
      WRITE (66, 9) ' NUEFM   = ', NUEFM, ' in m2/s'
      WRITE (66, *) '*****'
      WRITE (66, *) '          STOFFDATEN BEI WANDTEMPERATUR TW:'
      WRITE (66, *) '*****'
      WRITE (66, 9) ' ETAFW   = ', ETAFW, ' in kg/ms'
      WRITE (66, *) '*****'
      WRITE (66, *) '          DATEN BEI DER FLUESSIGKEITSTEMPERATUR TF:'
      WRITE (66, *) '*****'
      WRITE (66, 9) ' LAMF    = ', LAMF, ' in W/mK'
      WRITE (66, 9) ' ROF     = ', ROF, ' in kg/m3'
      WRITE (66, 9) ' ETAF    = ', ETAF, ' in kg/ms'
      WRITE (66, 9) ' CPF     = ', CPF, ' in J/kgK'
      WRITE (66, 9) ' PRF     = ', PRF, ' in (-)'
      WRITE (66, 9) ' HF      = ', HF, ' in J/kg'
      WRITE (66, *) '*****'
      WRITE (66, *) '          STOFFDATEN BEI 0.5 * (TW + TD):'
      WRITE (66, *) '*****'
      WRITE (66, 9) ' LAMDM   = ', LAMDM, ' in W/mK'
      WRITE (66, 9) ' RODM    = ', RODM, ' in kg/m3'
      WRITE (66, 9) ' ETADM   = ', ETADM, ' in kg/ms'
      WRITE (66, 9) ' CPDM    = ', CPDM, ' in J/kgK'
      WRITE (66, 9) ' PRDM    = ', PRDM, ' in (-)'
      WRITE (66, *) '*****'
      WRITE (66, *) '          STOFFDATEN BEI DER DAMPFTEMPORATUR TD:'
      WRITE (66, *) '*****'
      WRITE (66, 9) ' LAMD    = ', LAMD, ' in W/mK'
      WRITE (66, 9) ' ROD     = ', ROD, ' in kg/m3'
      WRITE (66, 9) ' ETAD    = ', ETAD, ' in kg/ms'
      WRITE (66, 9) ' CPD     = ', CPD, ' in J/kgK'
      WRITE (66, 9) ' PRD     = ', PRD, ' in (-)'
      WRITE (66, 9) ' HD      = ', HD, ' in J/kg'
      ENDIF
      FORMAT (A, TR2, D15.10, A, TR2)
      END

```



## Anhang D: Listing von HCINP

(Die Änderungen gegenüber der Originalversion sind markiert)

```

SUBROUTINE HCINP(TT,TTL,TTR,ATT,CP,HTC,HTCA,RHO,WLF,FWR1,FWR2,
*      FWR3,RV,SDE,SV,PCYLL,IVOLL,IVOLR,LAD,NGEO,ATTFAC,
*      HLENGT,NPTS,PLV,YARRAY,XARRAY,DHYL,
*      DHYR,TDNBL,TDNBR,TFBL,TFBR,MATL,
*      XMATL,FNAME,NOLAYS,TWBAL,TWBAR,TWBEL,TWBER,TL,
*      QTHRU,QHL,QHR,HCO,HCU,IALPH,ICHF,NHCOBJ,TANF)
CH+
CN      HCINP
CA      SHU
CM      AUH 87.06.22 (GCSM)
CM      DEI 87.11.10 (WRITE(IOUT,1110) MODIFIZIERT
CM      DEI 88.10.24 (GEOMETRIC POINT PLOTS)
CM      DEI 89.04.04 (ITIME CHANGED TO MTIME)
C*
CV      MOD.1.1A
C*
CP      - CONTROLS INPUT OF HC OBJECT DATA
CP      - DERIVES HCV SPECIFIC GEOMETRY AND MODEL DATA
CP      - PERFORMS ELEMENTARY CHECK OF INPUT AND DERIVED DATA
C*
CI      XMATL ( ) : NAMES OF MATERIALS
CI      TANF      : AVAILABLE CPU TIME AT START OF HECU
C*
CO      HTCA ( ) : SEE ROUTINE HECU
CO      LAD ( ) : SEE ROUTINE HECU
CO      PCYLL ( ) : SEE ROUTINE HECU
CO      NGEO ( ) : SEE ROUTINE HECU
CO      QHL ( ) : SEE ROUTINE HECU
CO      QHR ( ) : SEE ROUTINE HECU
C*
CU      DHYL ( ) : SEE ROUTINE HECU
CU      DHYR ( ) : SEE ROUTINE HECU
CU      HCO ( ) : SEE ROUTINE HECU
CU      HCU ( ) : SEE ROUTINE HECU
CU      HLENGT( ) : SEE ROUTINE HECU
CU      IALPH ( ) : SEE ROUTINE HECU
CU      ICHF ( ) : SEE ROUTINE HECU
CU      IVOLL ( ) : SEE ROUTINE HECU
CU      IVOLR ( ) : SEE ROUTINE HECU
CU      MATL ( ) : SEE ROUTINE HECU
CU      NOLAYS( ) : SEE ROUTINE HECU
CU      QTHRU ( ) : SEE ROUTINE HECU
CU      RV ( ) : SEE ROUTINE HECU
CU      TDNBL ( ) : SEE ROUTINE HECU
CU      TDNBR ( ) : SEE ROUTINE HECU
CU      TFBL ( ) : SEE ROUTINE HECU
CU      TFBR ( ) : SEE ROUTINE HECU
CU      TL ( ) : SEE ROUTINE HECU
CU      TTL ( ) : SEE ROUTINE HECU
CU      TTR ( ) : SEE ROUTINE HECU
CU      TWBAL ( ) : SEE ROUTINE HECU
CU      TWBAR ( ) : SEE ROUTINE HECU
CU      TWBEL ( ) : SEE ROUTINE HECU
CU      TWBER ( ) : SEE ROUTINE HECU
CU      FNAME ( ) : SEE COMMON VARIABLES DESCRIPTION
CU      ATT ( ) : SEE ROUTINE HECU
CU      CP ( ) : SEE ROUTINE HECU
CU      FWR1 ( ) : SEE ROUTINE HECU
CU      FWR2 ( ) : SEE ROUTINE HECU
CU      FWR3 ( ) : SEE ROUTINE HECU
CU      HTC ( ) : SEE ROUTINE HECU
CU      NPTS ( ) : SEE ROUTINE HECU

```

```

CU    PLV    ( ) : SEE ROUTINE HECU
CU    RHO    ( ) : SEE ROUTINE HECU
CU    SDE    ( ) : SEE ROUTINE HECU
CU    SV     ( ) : SEE ROUTINE HECU
CU    TT     ( ) : SEE ROUTINE HECU
CU    WLF    ( ) : SEE ROUTINE HECU
CU    XARRAY ( ) : SEE ROUTINE HECU
CU    YARRAY ( ) : SEE ROUTINE HECU
CU    NHCOBJ : SEE ROUTINE HECU
CU    ATTFAC ( ) : SEE ROUTINE HECU
C*
CF    MTIME   SGER8   SLER8   SPOLZ1
C*
CR    HCCHF   HCCOR   HCEQVL  HCFREE  HCFSET  HCFUNC  HCHEAT  HCMAT
CR    HCMSET  HCNOD   HCOBJ   HCOXIN  HCPRNT  HCTINI  HGRAPH  HPRPLO
CR    SERROR  SICPOS  SINTEG  SIPPOS
C*
CB    HINPUT
C*
CH-   INCLUDE (C)
      INCLUDE (CA)
      INCLUDE (CAARR)
      INCLUDE (CANWG2)
      INCLUDE (CANWC2)
      INCLUDE (CANW01)
      INCLUDE (CANW07)
      INCLUDE (CA002)
      INCLUDE (CDDR27)
      INCLUDE (CDGE20)
      INCLUDE (CDGE21)
      INCLUDE (CDGE23)
      INCLUDE (CDML01)
      INCLUDE (CDML02)
      INCLUDE (CDML14)
      INCLUDE (CDNW33)
      INCLUDE (CDNW34)
      INCLUDE (CDNW35)
      INCLUDE (CDNW36)
      INCLUDE (CDNW56)
      INCLUDE (CDNW59)
      INCLUDE (CDNW60)
      INCLUDE (CDPR19)
      INCLUDE (CDQ04)
      INCLUDE (CDTF56)
      INCLUDE (CGDR01)
      INCLUDE (CGDR03)
      INCLUDE (CHCDI)
      INCLUDE (CHCDIA)
      INCLUDE (CHCDI1)
      INCLUDE (CHCDI2)
      INCLUDE (CHCDI3)
      INCLUDE (CHCDM)
      INCLUDE (CHCDN4)
      INCLUDE (CHCDN5)
      INCLUDE (CHCDN6)
      INCLUDE (CHCDR)
      INCLUDE (CHCDRA)
      INCLUDE (CHCDRB)
      INCLUDE (CHCDRD)
      INCLUDE (CHCDRE)
      INCLUDE (CHCDRG)
      INCLUDE (CHCDRH)
      INCLUDE (CHCDRK)
      INCLUDE (CHCDRL)
      INCLUDE (CHCDRQ)
      INCLUDE (CHCDRS)
      INCLUDE (CHCDR1)
      INCLUDE (CHCDR2)
      INCLUDE (CHCDR3)
      INCLUDE (CHCDR4)
      INCLUDE (CHCDR5)

```

```

INCLUDE (CHCDR6)
INCLUDE (CHCDR7)
INCLUDE (CHCDR8)
INCLUDE (CHCDR9)
INCLUDE (CHCON2)
INCLUDE (CHCON3)
INCLUDE (CHEH01)
INCLUDE (CHEH09)
INCLUDE (CHI0)
INCLUDE (CHI1)
INCLUDE (CHMDOX)
INCLUDE (CHOXI1)
INCLUDE (CHOXR1)
INCLUDE (CHOXSI)
INCLUDE (CHP01)
INCLUDE (CHP02)
INCLUDE (CHP03)
INCLUDE (CHROXI)
INCLUDE (CHR0)
INCLUDE (CHSLG1)
INCLUDE (CHSLG4)
INCLUDE (CHZNXY)
INCLUDE (CHZRHO)
INCLUDE (CHZO2R)
INCLUDE (CHZRNA)
INCLUDE (CHZRVX)
INCLUDE (CHZRVY)
INCLUDE (ANDREA)
DIMENSION TT (LAYAL1),TTL ( L2IHV),TTR ( L2IHV),
* ATT (LAYAL1),ATTFAC(LAYAL1),
* CP (LAYAL1),HTC ( L2LAY),HTCA ( L2LAY),
* RHO (LAYAL1),WLF (LAYAL1),PCYLL ( IHV ),
* FWR1 (LAYAL2),FWR2 (LAYAL2),FWR3 (LAYAL2),
* RV (LAYAL2),SDE (LAYAL1),SV (LAYAL1),
* IVOLL ( IHV ),IVOLR ( IHV ),LAD ( IHV ),
* NCEO ( IHV ),TDNBR ( IHV ),
* TDNBL ( IHV ),TFBR ( IHV ),TFBL ( IHV ),
* DHYL ( IHV ),DHYR ( IHV ),HLENGT( IHV ),
* PLV (LAYAL1),YARRAY(MAXFUN),IALPH (LAYAL2),
* XARRAY(MAXFUN),
* NPTS (MFUNCT),MATL (LAY1P1),XMATL (MAXXMA),
* NOLAYS(L7IHV),FNAME (MFUNCT),
* TWBAL ( IHV ),TWBAR ( IHV ),TWBEL ( IHV ),
* TWBER ( IHV ),TL ( IHV ),ICHF ( L6IHV ),
* QTHRU ( IHV ),HCO ( IHV ),HCU ( IHV ),
* QHL (L2IHV),QHR (L2IHV)
C
DIMENSION ATT1 (000003),CPL (000003),HTCL (000004),
* RHOL (000003),WLFL (000003),XNAME (000009),
* PARTIT(000006),STEPE (000003),STEPV1(000003),
* STEPV2(000003),MATLAY(000003),TYP (000003),
* IALPHH(000004)
C
DIMENSION ITAB1(IHV), ITAB1(IHV)
C
LOGICAL SLER8,SGER8
CHARACTER*10 PSNAME,ADI,HCNAME,NONAME,FNAME
CHARACTER*14 MODEL(000003),MODE
CHARACTER*8 XNAME,TYP,XAREA,XLENTH,AORL,XMATL
CHARACTER*1 TABKEL,TABKER,BLANK,STERN
DOUBLE PRECISION QUOTH(1000), KPI
PARAMETER (KPI = 3.14159)
PARAMETER(DHC = 1.D-3)
C
EQUIVALENCE (MATLAY(1),N1),(MATLAY(2),N2),(MATLAY(3),N3)
C
SAVE NTABT
DATA XNAME /'R ','SDE ','VOL ','QLAYER','LAMBDA',
* 'HTC ','RHO ','CP ','TEMP '/,
* TYP /'PLATE ','HAWL CYL','FULL CYL'/,
* XAREA /'AREA '/,
* XLENTH/'LENGTH '/

```

```

DATA NERFRE/0/
DATA BLANK/' '/,STERN/'*'/
DATA MODEL(1)/' '/
DATA MODEL(2)/'FUEL ROD '/
DATA MODEL(3)/'HEAT EXCHANGER'/
DATA EPSILN /1.D-37/

C
C -- START VALUES
C
DATA IHVSUM/0/,LAYSUM/0/,NTABT/1/
DATA ADI/'ADIABAT '/
DATA NONAME/' '/
ISLAB = 0
IPOWER = 1
IF(NELRE .GT. 0) IPOWER = 0
IKSHC = IKS
IMATL = 1
IIIMAT = 0
IPRPLO = 0
KHCO = 0

C+AUH
LTABT=LTROD+3*NRODS
C-AUH
C
C -- LOOP OVER ALL HCOS
C
PSNAME = NONAME
1
CONTINUE
IF(KHCO .EQ. NHC OBJ) GO TO 999
IKSL = 1
IKSR = 1
ISITE = 0
KHCO = KHCO + 1

C
CALL SICPOS('HEATCOND ',ANAMH(KHCO),16,*999)
C
IF( IOPTHC .GE. 1)WRITE(IOUT, '( /1X,A,A10,3X,A,1I3,1X,102(1H-)/ )')
* 'HCO-NAME : ',ANAMH(KHCO),'#',KHCO
C
IF(KHCO .GT. NHC MAX) THEN
WRITE(IOUT,*)' NHC MAX=',NHC MAX
WRITE(IOUT,*)' CURRENT HCO-NUMBER = ',KHCO
CALL ERROR('HCINP ',16,1,'DIMENSION FOR HCO"S TO'//
* ' SMALL. GENERATE BLOCK DATA PROGRAM HBDAT WITH EXTENDED'//
* ' NHC MAX!',*998)
998
CONTINUE
ENDIF

C
C -- READ HEAT CONDUCTION OBJECT DATA
C
CALL HCOBJ(FNAME,KHCO,IPRPLO)
C
CALL SIPPOS(PSNAME,*999)
C
IF(PSNAME .EQ. 'HTCCORR ') THEN
CALL HCCOR
IHTCS = 0
C
IF( IOPTHC .GE. 1) THEN
WRITE(IOUT, '( ' PSEUDOKEYWORD HTCCORR ACTIV' )')
ENDIF
C
ELSE
IHTCS = 1
IHTC10 = 1
IHTC20 = 2
IHTC30 = 2
ENDIF
C
IF( ICHFO .EQ. 0) THEN
IF(PSNAME .NE. 'CHFREWET ')CALL SIPPOS(PSNAME,*999)
IF(PSNAME .NE. 'CHFREWET ') THEN

```

```

      CALL ERROR('HCINP      ',16,1,'PSEUDOKEYWORD CHFREWET NOT FOUND
* IN EXPECTED POSITION.',*1000)
      ENDIF
C
      CALL HCCHF
C
      ELSE
        DTNBL0 = 5.D-1
        DTNBR0 = 5.D-1
        DTRNL0 = 2.D0
        DTRNR0 = 2.D0
      ENDIF
C
      HCNAME = NONAME
      IF(PSNAME.NE.'HEATSOURCE'.AND.PSNAME.NE.'MATPROP') THEN
        CALL SIPPOS(PSNAME,*999)
      ENDIF
      IF(PSNAME.EQ.'HEATSOURCE') THEN
        IF(ICOMP0(KHCO).EQ.1) THEN
          CALL ERROR('HCINP      ',08,1,'PSEUDOKEYWORD HEATSOURCE MUST
* NOT BE USED FOR FUEL ROD MODEL OR ELECTRICAL HEATER MODEL',*988)
988      CONTINUE
          IQFUN0 = 0
          ATTL(1) = ZERO
          ATTL(2) = ZERO
          ATTL(3) = ZERO
        ELSE
          HCNAME = PSNAME
          CALL HCHEAT(KHCO,IQFUN0)
        ENDIF
      ELSE
        IQFUN0 = 0
        ATTL(1) = ZERO
        ATTL(2) = ZERO
        ATTL(3) = ZERO
      ENDIF
C
      IF(PSNAME.NE.'MATPROP')CALL SIPPOS(PSNAME,*999)
C
      IF(PSNAME.NE.'MATPROP') THEN
        CALL ERROR('HCINP      ',16,1,'PSEUDOKEYWORD MATPROP NOT FOUND
* IN EXPECTED POSITION.',*1000)
      ENDIF
      CALL HCMAT(FNAME,KHCO)
C
C -- CALCULATE ZR-OXIDATION, OPTIONAL
C
      PSNAME = NONAME
      CALL SIPPOS(PSNAME,*670)
C
      IF(PSNAME.EQ.'ZROXIDAT') THEN
C
        IF(IOPTHC.GE.1) THEN
          WRITE(IOUT,('' PSEUDOKEYWORD ZROXIDAT ACTIV''))
        ENDIF
C
        CALL HCOXIN(KHCO)
C
        IF(ASITLR(KHCO).EQ.'LEFT')  ISITE = 1
        IF(ASITLR(KHCO).EQ.'RIGHT') ISITE = 2
        IF(ASITLR(KHCO).EQ.'BOTH')  ISITE = 3
C
      ENDIF
C
670  CONTINUE
C
C -- DERIVED QUANTITIES
C
      IF(AOLH(KHCO).EQ.ADI) SBOLH = DABS(SBOLH)
      IF(AORH(KHCO).EQ.ADI) SBORH = DABS(SBORH)
      IF(SBOLH.LT.ZERO.AND.SBORH.LT.ZERO) THEN
        WRITE(IOUT,*)' ERROR MESSAGE ERROR MESSAGE ERROR MESSAGE'

```

```

        WRITE(IOUT,*)' '
        WRITE(IOUT,*)' CHECK YOUR INPUT: SBOLH,SBORH'
        CALL SERROR('HCINP      ',16,1,'SBOLH AND SBORH ARE LESS'//
*      ' THAN ZERO.',*340)
    ENDIF
C
    IF(IOPTHC .GE. 1) THEN
        WRITE(IOUT,*)' '
        WRITE(IOUT,*)'HCO-DESCRIPTION:'
    ENDIF
C
    IF(SBOLH .GE. ZERO .AND. SBORH .GE. ZERO) THEN
        ITH = 1
C
        IF(IOPTHC .GE. 1) THEN
            WRITE(IOUT,'(A,115)') ' NO TEMPERATURE SIGNAL USED. ITH= ',
*          ITH
        ENDIF
C
    ENDIF
C
    IF(SBOLH .LT. ZERO .AND. SBORH .GE. ZERO) THEN
        ITH = 2
        SBOLH = DABS(SBOLH)
        SEOLH = 1.D30
        NTABE = NTABT + IRHO + NMPROP - 1
        NTABS = IRHO + NMPROP
C
        DO 47 J = NTABS,NTABE
C AUH      IF(FNAME(J) .EQ. AOLH(KHCO)) THEN
            IF(FNAME(J) .EQ. AOLH(KHCO)      ) THEN
                NKSL = NTABS - J - 1
                GO TO 46
            ENDIF
47      CONTINUE
C
C AUH      FNAME(NTABE) = AOLH(KHCO)
            FNAME(NTABE) = AOLH(KHCO)
            NKSL = - NTABT
C+AUH
            ACTNAM(LTABT+NTABT-1)=AOLH(KHCO)
C-AUH
            IF(NTABT .GT. NTHIST) THEN
                WRITE(IOUT,*)' ERROR MESSAGE ERROR MESSAGE ERROR MESSAGE'
                WRITE(IOUT,*)'HCO NAME: ',ANAMH(KHCO)
                WRITE(IOUT,*)'TEMPERATURE SIGNAL NAMES FOUND:'
                WRITE(IOUT,'(1X,10A12)')(FNAME(J),J=NTABS,NTABE)
                WRITE(IOUT,*)'NTHIST = ',NTHIST
                CALL SERROR('HCINP      ',16,1,'TOO MANY TEMPERATURE SIGNALS.'
*              '// ' CHECK NTHIST AND/OR YOUR SIGNAL NAMES.',*340)
            ENDIF
C
            NTABT = NTABT + 1
46      CONTINUE
C
            IF(IOPTHC .GE. 1) THEN
                WRITE(IOUT,'(A,115)')' TEMPERATURE SIGNAL ON THE LEFT SIDE'//
*              ',OBJECT ON THE RIGHT SIDE. ITH= ',ITH
            ENDIF
C
        ENDIF
C
    IF(SBOLH .GE. ZERO .AND. SBORH .LT. ZERO) THEN
        ITH = 3
        SBORH = DABS(SBORH)
        SEORH = 1.D30
        NTABE = NTABT + IRHO + NMPROP - 1
        NTABS = IRHO + NMPROP
C
        DO 57 J = NTABS,NTABE
            IF(FNAME(J) .EQ. AORH(KHCO)      ) THEN
                NKSR = NTABS - J - 1

```

```

        GO TO 56
    ENDIF
57    CONTINUE
C
    FNAME(NTABE) = AORH(KHCO)
    NKSR = - NTABT
C+AUH
    ACTNAM(LTABT+NTABT-1)=AORH(KHCO)
C-AUH
    IF(NTABT .GT. NTHIST) THEN
        WRITE(IOUT,*)' ERROR MESSAGE ERROR MESSAGE ERROR MESSAGE'
        WRITE(IOUT,*)'HCO NAME: ',ANAMH(KHCO)
        WRITE(IOUT,*)'TEMPERATURE SIGNAL NAMES FOUND:'
        WRITE(IOUT, '(1X,10A12)')(FNAME(J),J=NTABS,NTABE)
        WRITE(IOUT,*)'NTHIST = ',NTHIST
        CALL SERROR('HCINP      ',16,1,'TOO MANY TEMPERATURE SIGNALS.'
*      // ' CHECK NTHIST AND/OR YOUR SIGNAL NAMES.',*340)
    ENDIF
C
    NTABT = NTABT + 1
56    CONTINUE
C
    IF(IOPTHC .GE. 1) THEN
        WRITE(IOUT, '(A,1I5)')' TEMPERATURE SIGNAL ON THE RIGHT SID'//
*      'E, OBJECT ON THE LEFT SIDE. ITH= ',ITH
    ENDIF
C
    ENDIF
C
C -- FIND TFO-KEYWORD AND INDEX,LEFT
C
    IF(ITH .EQ. 2) GO TO 22
    K = 0
    IF(AOLH(KHCO) .EQ. ADI) THEN
        IF(IOPTHC .GE. 1)WRITE(IOUT, '(' LEFT SIDE IS ADIABATIC.'))'
        SBOLH = 0.0D0
        SEOLH = 0.0D0
        IKSL = 0
        GO TO 22
    ENDIF
420    CONTINUE
    K = K + 1
    IF(K .GT. NOBJ) THEN
        WRITE(IOUT,*)' ERROR MESSAGE ERROR MESSAGE ERROR MESSAGE'
        WRITE(IOUT,*)'HCO NAME: ',ANAMH(KHCO)
        WRITE(IOUT,*)'KEYWORD : ',AOLH(KHCO)
        WRITE(IOUT,*)'NOBJ      : ',NOBJ
        CALL SERROR('HCINP      ',16,1,'KEYWORD(LEFT) NOT FOUND',*997)
997    CONTINUE
    ENDIF
    IF(AOLH(KHCO) .NE. ANAMO(K)) GO TO 420
    KL = K
    IJLOKL = IJLO(KL)
    IJROKL = IJRO(KL)
C
C -- FIND TFO-KEYWORD AND INDEX,RIGHT
C
22    CONTINUE
    IF(ITH .EQ. 3) GO TO 23
    K = 0
    IF(AORH(KHCO) .EQ. ADI) THEN
        IF(IOPTHC .GE. 1)WRITE(IOUT, '(' RIGHT SIDE IS ADIABATIC.'))'
        SBORH = 0.0D0
        SEORH = 0.0D0
        IKSR = 0
        GO TO 23
    ENDIF
21    CONTINUE
    K = K + 1
C
    IF(K .GT. NOBJ) THEN
        WRITE(IOUT,*)' ERROR MESSAGE ERROR MESSAGE ERROR MESSAGE'

```

```

        WRITE(IOUT,*)'HCO NAME: ',ANAMH(KHCO)
        WRITE(IOUT,*)'KEYWORD : ',AORH(KHCO)
        WRITE(IOUT,*)'NOBJ      : ',NOBJ
        CALL ERROR('HCINP      ',16,1,'KEYWORD(RIGHT) NOT FOUND',*996)
996      CONTINUE
      ENDIF
C
      IF(AORH(KHCO) .NE. ANAMO(K)) GO TO 21
      KR = K
      IJROKR = IJRO(KR)
      IJLOKR = IJLO(KR)
23      CONTINUE
      IF(IOPTHC .GE. 1) THEN
        IF(IKSL .NE. 0 .AND. ITH .NE. 2)
          *WRITE(IOUT, '(1X,A,1A10,A,1I5)') 'TFO-KEYWORD(LEFT) : ',ANAMO(KL),
          *' #OBJECT : ',KL
          IF(IKSR .NE. 0 .AND. ITH .NE. 3)
            *WRITE(IOUT, '(1X,A,1A10,A,1I5)') 'TFO-KEYWORD(RIGHT): ',ANAMO(KR),
            *' #OBJECT : ',KR
          IF(IKSL .NE. 0 .AND. ITH .EQ. 2)
            *WRITE(IOUT, '(1X,A,1A10)')      'SIGNAL NAME(LEFT)  : ',AOLH(KHCO)
            IF(IKSL .NE. 0 .AND. ITH .EQ. 3)
              *WRITE(IOUT, '(1X,A,1A10)')      'SIGNAL NAME(RIGHT) : ',AORH(KHCO)
          ENDIF
C
        IF(IOPTHC .GE. 1 .AND. ISITE .GT. 0) THEN
          IF(ISITE .EQ. 1) WRITE(IOUT,*)'OXIDATION ON LEFT SURFACE'//
          *      ' IS POSSIBLE'
          IF(ISITE .EQ. 2) WRITE(IOUT,*)'OXIDATION ON RIGHT SURFACE'//
          *      ' IS POSSIBLE'
          IF(ISITE .EQ. 3) WRITE(IOUT,*)'OXIDATION ON BOTH SURFACES'//
          *      ' IS POSSIBLE'
        ENDIF
C
C -- CALCULATE HECU NODALISATION
C
      IJLA = 1
      IJLE = 1
C
      IF(ITH .NE. 2 .AND. IKSL .NE. 0) THEN
C
        IF(ITYPO(KL) .EQ. 22 .OR. ITYPO(KL) .EQ. 30) THEN
          WRITE(IOUT,*)'HCO NAME: ',ANAMH(KHCO)
          WRITE(IOUT,*)' ITYPO = ',ITYPO(KL)
          CALL ERROR('HCINP      ',16,1,'ITYPO MUST NOT'//
          *      'BE 22 OR 30! ',*1000)
        ENDIF
C
        IF((ITYPO(KL) .GE. 10 .AND. ICMPO(KL) .NE. 3) .OR.
          *      IART (KL) .EQ. 14 .OR. IART (KL) .EQ. 17) THEN
          IJLA = 1
          IJLE = IJROKL - IJLOKL + IJLA
C
          IF(IJLE .GT. NHELPM) THEN
            WRITE(IOUT,*)'NHELPM = ',NHELPM
            CALL ERROR('HCINP      ',16,1,'DIMENSION ERROR.'//
            *      ' GENERATE BLOCK DATA PROGRAM HBDAT WITH EXTENDED'//
            *      ' NHELPM! ',*1000)
          ENDIF
C
          J = IJLA
          DO 500 I = IJLOKL , IJROKL
            SOJLR(J) = SOJ(I)
            J = J + 1
500      CONTINUE
C
        ELSE
          IJLA = 1
          IJLE = 2
          SOJLR(IJLA) = SOLI(IILO(KL))
          SOJLR(IJLE) = SORI(IILO(KL))
        ENDIF

```



```

C      ENDIF
C
C      IF(ITH .NE. 3 .AND. IKSR .NE. 0) THEN
C
C          IF(ITYP0(KR) .EQ. 22 .OR. ITYP0(KR) .EQ. 30) THEN
C              WRITE(IOUT,*)'HCO NAME: ',ANAMH(KHCO)
C              WRITE(IOUT,*)' ITYP0 = ',ITYP0(KR)
C              CALL SERR0R('HCINP      ',16,1,'ITYP0 MUST NOT'//
*              'BE 22 OR 30! ',*1000)
C              ENDIF
C
C          IF((ITYP0(KR).GE.10.AND.ICMPO(KR).NE.3).OR.
*          IART (KR).EQ.14.OR. IART (KR).EQ.17) THEN
C              IJRA = IJLE + 1
C              IJRE = IJROKR - IJLOKR + IJRA
C
C              IF(IJRE .GT. NHELPM) THEN
C                  WRITE(IOUT,*)'NHELPM = ',NHELPM
C                  CALL SERR0R('HCINP      ',16,1,'DIMENSION ERROR.'//
*                  ' GENERATE BLOCK DATA PROGRAM HBDAT WITH EXTENDED'//
*                  ' NHELPM! ',*1000)
C                  ENDIF
C
C              J = IJRA
C              DO 510 I = IJLOKR , IJROKR
C                  SOJLR(J) = SOJ(I)
C                  J = J + 1
510      CONTINUE
C
C              ELSE
C                  IJRA = IJLE + 1
C                  IJRE = IJRA + 1
C                  SOJLR(IJRA) = SOLI(IIRO(KR))
C                  SOJLR(IJRE) = SORI(IIRO(KR))
C              ENDIF
C
C      ENDIF
C
C      NHCV = 0
C      CALL HCNOD(KL,KR,IJLA,IJLE,IJRA,IJRE,NHCV,ITH,NZAEHL,KHCO,1)
C      NKHCO(KHCO) = NHCV
C
C      IF(NHCV .GT. NHELPM) THEN
C          WRITE(IOUT,*)'NHELPM = ',NHELPM
C          CALL SERR0R('HCINP      ',16,1,'DIMENSION ERROR.'//
*          ' GENERATE BLOCK DATA PROGRAM HBDAT WITH EXTENDED'//
*          ' NHELPM! ',*1000)
C      ENDIF
C
C      IF(IOPTH0 .GE. 1) THEN
C          WRITE(IOUT,*)'NUMBER OF HCVS IN THIS HCO: ',NHCV
C      ENDIF
C
C      IF(SOH(NHCV) .GT. ZERO) THEN
C          FACS0H = SG0(NCARD1) / SOH(NHCV)
C          IF(DABS(FACS0H - ONE) .LE. 1.0D-3) FACS0H = ONE
C      ENDIF
C
C      IF(KHCO .EQ. 1) THEN
C          IA = 1
C          IE = NHCV
C      ELSE
C          IA = IE + 1
C          IE = IE + NHCV
C      ENDIF
C
C      CDEI+ FUER ORTSPLOTS
C          IAHO(KHCO) = IA
C      CDEI-
C
C      INHV = 0

```

```

DO 410 NHV=IA,IE
C
IF(IOPTHC .GE. 3) THEN
  WRITE(IOUT, '(1X,A,1I5/1X,13(1H=)/)') 'HCV-NR.: ',NHV
ENDIF
C
INHVB = INHVB + 1
ISLAB = ISLAB + 1
LAY1SV = 1
LAY2SV = 1
NOLAYS(NHV) = N10 + N20 + N30
LAYSUM = LAYSUM + NOLAYS(NHV)
NLAY = NOLAYS(NHV)
NLAYM1 = NLAY - 1
IFH = 3 * (NHV - 1) + 1
ISH = IFH + 2
IFQ = 6 * (NHV - 1) + 1
ISQ = IFQ + 5
N2 = N20
N3 = N30
C
ICHFS = ICHFO
DO 7 I=1,6
7 IF(ICHFO .EQ. 1) ICHF(IFQ+I-1)=IHDATA(I+3)
DO 8 I=1,6
8 IF(ICHFO .EQ. 2) ICHF(IFQ+I-1)=IHDATA(I+9)
DO 9 I=1,6
9 IF(ICHFO .EQ.-1) ICHF(IFQ+I-1)=IHDATA(I+15)
C
TFBL(NHV) = DTNBL0
TFBR(NHV) = DTNBR0
TWBEL(NHV) = DTRNLO
TWBER(NHV) = DTRNR0
C
IF(ICHFO .EQ. 0) THEN
  ICHF(IFQ) = ICHF10
  ICHF(IFQ+1)= ICHF20
  ICHF(IFQ+2)= ICHF30
  ICHF(IFQ+3)= ICHF40
  ICHF(IFQ+4)= ICHF50
  ICHF(IFQ+5)= ICHF60
C
IF(ICHF(IFQ) .EQ. -1) ICHF(IFQ+1)=1
C
IF(ICHF(IFQ+3) .EQ. -1) ICHF(IFQ+4)=1
C
IF(ICHF(IFQ) .EQ. -1 .OR. ICHF(IFQ+1) .EQ. 1) THEN
  TDNBL(NHV) = TDNBL0
  TDNBR(NHV) = TDNBR0
C
IF(IOPTHC .GE. 3) THEN
  WRITE(IOUT, '(1X,1P,A,/1X,4D12.4)') 'TDNBL,TDNBR,TFBL,TFBR',
*   TDNBL(NHV),TDNBR(NHV),TFBL(NHV),TFBR(NHV)
  ENDIF
C
ENDIF
C
IF(ICHF(IFQ+3) .EQ. -1 .OR. ICHF(IFQ+4) .EQ. 1) THEN
  TWBAL(NHV) = TWBAL0
  TWBAR(NHV) = TWBAR0
C
IF(IOPTHC .GE. 3) THEN
  WRITE(IOUT, '(1X,1P,A,/1X,4D12.4)') 'TWBAL,TWBAR,TWBEL,TEBER',
*   TWBAL(NHV),TWBAR(NHV),TWBEL(NHV),TWBER(NHV)
  ENDIF
C
ENDIF
C
ENDIF
C
IF(IOPTHC .GE. 3) THEN
  WRITE(IOUT, '(1X,A,6I5)') 'ICHF(I)=', (ICHF(I),I=IFQ,ISQ)

```

```

      ENDIF
C
      IHTC(IFH) = IHTC10
      IHTC(IFH+1) = IHTC20
      IHTC(IFH+2) = IHTC30
C
      IF(IOPTHC .GE. 3) THEN
        WRITE(IOUT,'(1X,A,3I5)') 'IHTC(I)=',(IHTC(I),I=IFH,ISH)
      ENDIF
C
      IF(IKSL .EQ. 0) THEN
        NKSL = 0
        GO TO 470
      ENDIF
      IF(ITH .EQ. 2) GO TO 470
      IF(IKSL .NE. 0 .AND. SBOLH .LT. SEOLH) THEN
        J = 0
        DO 430 I = IJLA,IJLE
C
          IF(SGER8(SOJLR(I),SOH(INHV) + SBOLH) .OR.
*           SGER8(SOJLR(I),SEOLH)) THEN
            NKSL = IILO(KL) + J - 1
            GO TO 31
          ENDIF
C
          J = J + 1
430        CONTINUE
          NKSL = IIRO(KL)
31        CONTINUE
        ELSE
          J = 0
          SBLH = SOJLR(IJLE)-SBOLH
          DO 440 I = IJLA,IJLE
C
            IF(SGER8(SOJLR(IJLE) - SOJLR(IJLE-J),SBLH + SOH(INHV)) .OR.
*             SGER8(SOJLR(IJLE) - SOJLR(IJLE-J),SBLH+SBOLH-SEOLH)) THEN
              NKSL = IIRO(KL) - J + 1
              GO TO 41
            ENDIF
C
            J = J + 1
440          CONTINUE
          NKSL = IIRO(KL)
C
41          CONTINUE
        ENDIF
C
        IF(NKSL .LT. IILO(KL) .OR. NKSL .GT. IIRO(KL)) THEN
          WRITE(IOUT,*)'HCO NAME ',ANAMH(KHCO)
          WRITE(IOUT,*)'NKSL = ',NKSL
          WRITE(IOUT,*)'CHECK SBOLH,SEOLH'
          CALL SERROR('HCINP ',12,1,'NODALISATION ERROR.THIS VOLUME'//
* ' IS A BOUNDARY VOLUME !',*470)
        ENDIF
C
470      CONTINUE
C
      IF(IKSR .EQ. 0) THEN
        NKSR = 0
        GO TO 480
      ENDIF
      IF(ITH .EQ. 3) GO TO 480
      IF(IKSR .NE. 0 .AND. SBORH .LT. SEORH) THEN
        J = 0
        DO 450 I = IJRA,IJRE
C
          IF(SGER8(SOJLR(I),SOH(INHV)+SBORH) .OR.
*           SGER8(SOJLR(I),SEORH)) THEN
            NKSR = IILO(KR) + J - 1
            GO TO 51
          ENDIF
C

```

```

      J = J + 1
450    CONTINUE
      NKSR = IIRO(KR)
51     CONTINUE
      ELSE
        J = 0
        SBRH = SOJLR(IJRE)-SBORH
        DO 460 I = IJRA,IJRE
C
          IF(SGER8(SOJLR(IJRE) - SOJLR(IJRE-J),SBRH + SOH(INHV)) .OR.
*         SGER8(SOJLR(IJRE) - SOJLR(IJRE-J),SBRH+SBORH-SEORH)) THEN
            NKSR = IIRO(KR) - J + 1
            GO TO 61
          ENDIF
C
          J = J + 1
460    CONTINUE
      NKSR = IILO(KR)
C
61     CONTINUE
      ENDIF
C
      IF(NKSR .LT. IILO(KR) .OR. NKSR .GT. IIRO(KR)) THEN
        WRITE(IOUT,*)'HCO NAME ',ANAMH(KHCO)
        WRITE(IOUT,*)'NKSR = ',NKSR
        WRITE(IOUT,*)'CHECK SBORH,SEORH'
        CALL SERROR('HCINP      ',12,1,'NODALISATION ERROR.THIS VOLUME'//
*       ' IS A BOUNDARY VOLUME !',*480)
      ENDIF
C
480    CONTINUE
C
C -- IF TFV IS RIGHTANGLE TOF HCV
C
      IF((NKSL .GT. 0 .AND. DABS(SEOLH-SBOLH) .LT. 1.D-5) .OR.
*     (NKSR .GT. 0 .AND. DABS(SEORH-SBORH) .LT. 1.D-5)) THEN
C
        SOH(INHV) = SG0(NCARD1) / DFLOAT(NIHCO) * DFLOAT(INHV)
      ELSE
        SOH(INHV) = SOH(INHV) * FACS0H
C
        IF(FACS0H .NE. ONE .AND. IOPTHC .GE. 3) THEN
          WRITE(IOUT,'(1X,1P,A,1D12.4)')'LENGTH COORDINATES OF HCV'//
*         ' ARE PROJECTED WITH THE FACTOR : ',FACS0H
        ENDIF
C
      ENDIF
C
      IF(IOPTHC .GE. 3) THEN
        WRITE(IOUT,'(1X,1P,A,1D12.4)')'SOH = ',SOH(INHV)
        WRITE(IOUT,'(1X,A,1I5,2X,A,1I5)')'NKSL=',NKSL,'NKSR=',NKSR
      ENDIF
C
      NGEOR = IGEOO
C
      TL(NHV) = TLO
      FPAR(NHV) = FPARH
C
      DO 300 I = 1,NCARD1
        IF(SGER8(SG0(I),SOH(INHV))) GO TO 301
300    CONTINUE
      IGRENZ = NCARD1
      GO TO 302
301    CONTINUE
      IGRENZ = I
302    CONTINUE
C
      IF(NCARD1 .GT. 1 .AND. ZO(IGRENZ-1) .GT. ZO(IGRENZ)) THEN
        IF(INHV .EQ. 1) THEN
          HCO(NHV) = SPOLZ1(1,NCARD1,EPSILN,SG0,ZO,1)
        ELSE
          HCO(NHV) = SPOLZ1(1,NCARD1,SOH(INHV-1),SG0,ZO,1)

```

```

      ENDIF
C      HCU(NHV) = SPOLZ1(1,NCARD1,SOH(INHV),SG0,Z0,1)
      ELSE
        IF(INHV.EQ.1) THEN
          HCU(NHV) = SPOLZ1(1,NCARD1,EPSILN,SG0,Z0,1)
        ELSE
          HCU(NHV) = SPOLZ1(1,NCARD1,SOH(INHV-1),SG0,Z0,1)
        ENDIF
C      HCO(NHV) = SPOLZ1(1,NCARD1,SOH(INHV),SG0,Z0,1)
      ENDIF
C      NMIXL = 0
      DO 85 I = 1,NZAEHL
C      IF(NKSL.GT.0) THEN
        NL = NKSL
      ELSE
        NL = 1
      ENDIF
C      IF(NKSR.GT.0) THEN
        NR = NKSR
      ELSE
        NR = 1
      ENDIF
C      IF((IMLK(NL).EQ.1.AND.NKSL.GT.0).OR.
      * (IMLK(NR).EQ.1.AND.NKSR.GT.0)) THEN
        IF(IMLK(NL).EQ.1.AND.NKSL.GT.0) ZGEO = ZML(IML(NL))
        IF(IMLK(NR).EQ.1.AND.NKSR.GT.0) ZGEO = ZML(IML(NR))
        IF(DABS(HCO(NHV)-ZGEO).LT.DHC/2.D0) THEN
          HCO(NHV) = HCO(NHV)-DHC
          NMIXL = 1
        ENDIF
        IF(DABS(HCU(NHV)-ZGEO).LT.DHC/2.D0) THEN
          HCU(NHV) = HCU(NHV)+DHC
          NMIXL = 1
        ENDIF
      ENDIF
85    CONTINUE
C      IF(IOPTHC.GE.3)WRITE(IOUT,'(1X,1P,A,1D14.6,3X,A,1D14.6)')
      *'HCO=',HCO(NHV),'HCU=',HCU(NHV)
C      IF(INHV.EQ.1) THEN
        SOL = 0.D0
        SOR = SOH(INHV)
      ELSE
        SOL = SOH(INHV-1)
        SOR = SOH(INHV)
      ENDIF
C      IF(IGEO0.EQ.0.OR.IGEO0.EQ.3) THEN
        PARTIT(1) = ZERO
      ELSE
        CALL SAVER (SOL,SOR,SG0,DS10,PARTIT(1),NCARD1,1)
        PARTIT(1) = 5.D-1*PARTIT(1)
      ENDIF
      CALL SAVER (SOL,SOR,SG0,DS10,PARTIT(2),NCARD1,1)
      CALL SAVER (SOL,SOR,SG0,GAP10,PARTIT(3),NCARD1,1)
      CALL SAVER (SOL,SOR,SG0,DS20,PARTIT(4),NCARD1,1)
      CALL SAVER (SOL,SOR,SG0,GAP20,PARTIT(5),NCARD1,1)
      CALL SAVER (SOL,SOR,SG0,DS30,PARTIT(6),NCARD1,1)
      IF(IOPTHC.GE.3)
      *WRITE(IOUT,'(1X,1P,A,6D12.4)')'PARTIT(1...6)',(PARTIT(I),I=1,6)
C      SO = ZERO
      SU = SOH(INHV)
      IF(INHV.NE.1) SO = SOH(INHV-1)
      IF(IGEO0.EQ.0.OR.IGEO0.EQ.3) THEN

```

```

      CALL SINTEG(SO,SOH(INHV),SG0,DIO,F,NCARD1,1)
      AORXL = F * FPARH
    ELSE
      AORXL = FPARH * (SOH(INHV) - SO)
    ENDIF
C
    HLENGT(NHV) = SOH(INHV) - SO
    HLQUER      = SOH(NHCV) / DFLOAT(NHCV)
    IF(HLENGT(NHV)/HLQUER .LT. 5.0D-2) THEN
      WRITE(IOUT,'(1X,1P,A,1D12.4)') 'THE AVERAGE LENGTH OF A HCV '//
*   'IN THIS HCO IS : ',HLQUER
      WRITE(IOUT,'(1X,1P,A,1D12.4)') 'THE LENGTH OF THE CURRENT HC'//
*   'V IS : ',HLENGT(NHV)
      WRITE(IOUT,*) 'THE LENGTH OF THE CURRENT HCV IS LESS THAN 5% '//
*   'OF THE AVERAGE LENGTH !'
      WRITE(IOUT,*) 'CONTROL WHETHER THE SMALL HCV HAS TO BE '//
*   'CONSIDERED !'
      CALL ERROR('HCINP ',04,1,'W A R N I N G : THE VOLUME'//
*   ' LENGTH IS VERY SMALL !',*341)
341  CONTINUE
    ENDIF
C
    IF(NKSL .LE. 0) THEN
      DHYL(NHV) = 0.D0
    ELSE
      DHYL(NHV) = DHYI(NKSL)
    ENDIF
C
    IF(NKSR .LE. 0) THEN
      DHYR(NHV) = 0.D0
    ELSE
      DHYR(NHV) = DHYI(NKSR)
    ENDIF
C
    IF(IOPTHC .GE. 3)WRITE(IOUT,'(1X,1P,2(A,1D12.4,3X,A,1D12.4,3X))')
* 'AORXL=',AORXL,'HLENGT=',HLENGT(NHV),
* 'DHYL=',DHYL(NHV),'DHYR=',DHYR(NHV)
C
    DO 10 I=1,4
      IALPHH(I) = IALPH0(I)
10  CONTINUE
C
    DO 12 J =1,4
C
      L = 0
      N = 1
      JE = NCARD3 - 1 + J
      DO 11 I = J,JE
        SOHELP(N) = HTCL0(I+L)
        L = L + 3
        N = N + 1
11  CONTINUE
      IF(N .GT. NHELPM) THEN
        WRITE(IOUT,*) 'NHELPM = ',NHELPM
        CALL ERROR('HCINP ',16,1,'DIMENSION ERROR.'//
*   ' GENERATE BLOCK DATA PROGRAM HBDAT WITH EXTENDED'//
*   ' NHELPM!',*1000)
      ENDIF
C
      HTCL(J) = (SPOLZ1(1,NCARD3,SO,SH0,SOHELP,1) +
*   SPOLZ1(1,NCARD3,SOH(INHV),SH0,SOHELP,1))/2.D0
12  CONTINUE
      IF(IOPTHC .GE. 3)WRITE(IOUT,'(1X,1P,2(A,1D12.4,3X,A,1D12.4,3X))')
* 'HTCL1=',HTCL(1),'HTCL2=',HTCL(2),
* 'HTCL3=',HTCL(3),'HTCL4=',HTCL(4)
C
      CALL SINTEG(SO,SOH(INHV),SH0,QTHRU0,F,NCARD3,1)
      QTHRU(NHV) = F * FPARH
C
      IF(IOPTHC .GE. 3)WRITE(IOUT,'(1X,1P,A,1D12.4)') 'QTHRU=',QTHRU(NHV)
C
      IF(HCNAME .EQ. 'HEATSOURCE') THEN
        IQFUNC(NHV) = IQFUN0

```

```

C
C      IF(IGEOO .EQ. 0 .OR. IGEOO .EQ. 3) THEN
C
C          V1 = PARTIT(2)*AORXL/FPARH
C          V2 = PARTIT(4)*AORXL/FPARH
C          V3 = PARTIT(6)*AORXL/FPARH
C
C      ELSE
C
C          R1 = PARTIT(1)
C          R2 = R1 + PARTIT(2)
C          V1 = PII*HLENGT(NHV)*(R2**2 - R1**2)
C
C          R1 = R2 + PARTIT(3)
C          R2 = R1 + PARTIT(4)
C          V2 = PII*HLENGT(NHV)*(R2**2 - R1**2)
C
C          R1 = R2 + PARTIT(5)
C          R2 = R1 + PARTIT(6)
C          V3 = PII*HLENGT(NHV)*(R2**2 - R1**2)
C
C      ENDIF
C
C      V4 = V1
C      IF(V1 .LT. EPSILN) V1 = ONE
C      V5 = V2
C      IF(V2 .LT. EPSILN) V2 = ONE
C      V6 = V3
C      IF(V3 .LT. EPSILN) V3 = ONE
C
C      CALL SINTEG(SO,S0H(INHV),SA0,ATTL10,F,NCARD4,1)
C      ATTL(1) = F/V1
C      CALL SINTEG(SO,S0H(INHV),SA0,ATTL20,F,NCARD4,1)
C      ATTL(2) = F/V2
C      CALL SINTEG(SO,S0H(INHV),SA0,ATTL30,F,NCARD4,1)
C      ATTL(3) = F/V3
C      IF(IOPTHC .GE. 3) THEN
C          WRITE(IOUT,'(1X,1P,3(A,1D12.4,3X))')
C      * 'ATTL1=',ATTL(1),'ATTL2=',ATTL(2),'ATTL3=',ATTL(3)
C          WRITE(IOUT,'(1X,1P,3(A,1D12.4,3X))')
C      * 'V1=',V4,'V2=',V5,'V3=',V6
C      ENDIF
C
C      ENDIF
C
C      MATL1 = MATL10
C      MATL2 = MATL20
C      MATL3 = MATL30
C
C      IF(IOPTHC .GE. 3)WRITE(IOUT,'(1X,3(A,1I5,3X))')
C      * 'MATL1=',MATL1,'MATL2=',MATL2,'MATL3=',MATL3
C
C      IF(MATL1 .EQ. 0) THEN
C          WLFL(1) = WLF10
C          CPL(1) = CPL10
C          RHOL(1) = RHOL10
C
C          IF (WLFL(1).EQ.ZERO.OR.CPL(1).EQ.ZERO .OR. RHOL(1).EQ.ZERO) THEN
C              IIIMAT=IIIMAT+1
C          ENDIF
C
C          IF(IOPTHC .GE. 3)WRITE(IOUT,'(1X,1P,3(A,1D12.4,3X))')
C      * 'WLFL1=',WLFL(1),'CPL1=',CPL(1),'RHOL1=',RHOL(1)
C
C      ELSE
C          WLFL(1) = 0.D0
C          CPL(1) = 0.D0
C          RHOL(1) = 0.D0
C      ENDIF
C
C      IF(MATL2 .EQ. 0 .AND. N2 .GT. 0) THEN
C          WLFL(2) = WLF20

```

```

      CPL(2) = CPL20
      RHOL(2) = RHOL20
C
      IF (WLFL(2).EQ.ZERO.OR.CPL(2).EQ.ZERO .OR. RHOL(2).EQ.ZERO) THEN
        IIIMAT=IIIMAT+1
      ENDIF
C
      IF(IOPTHC .GE. 3)WRITE(IOUT,'(1X,1P,3(A,1D12.4,3X))')
*   'WLFL2=',WLFL(2),'CPL2=',CPL(2),'RHOL2=',RHOL(2)
C
      ELSE
        WLFL(2) = 0.D0
        CPL(2) = 0.D0
        RHOL(2) = 0.D0
      ENDIF
C
      IF(MATL3 .EQ. 0 .AND. N3 .GT. 0) THEN
        WFL(3) = WFL30
        CPL(3) = CPL30
        RHOL(3) = RHOL30
C
        IF (WLFL(3).EQ.ZERO.OR.CPL(3).EQ.ZERO .OR. RHOL(3).EQ.ZERO) THEN
          IIIMAT=IIIMAT+1
        ENDIF
C
        IF(IOPTHC .GE. 3)WRITE(IOUT,'(1X,1P,3(A,1D12.4,3X))')
*   'WLFL3=',WLFL(3),'CPL3=',CPL(3),'RHOL3=',RHOL(3)
C
        ELSE
          WLFL(3) = 0.D0
          CPL(3) = 0.D0
          RHOL(3) = 0.D0
        ENDIF
C
        AORX(NHV) = AORXL
C
C -- START VALUES OF OXIDE THICKNESS
C
      IF(IOXTF .GT. 0) THEN
C
        IF(ISITE .EQ. 1 .OR. ISITE .EQ. 3) THEN
          HROXID(NHV) = ZROX0 * FPARH
          HROXID(2*IHV+NHV) = ZROX0 * FPARH
        ENDIF
        IF(ISITE .EQ. 2 .OR. ISITE .EQ. 3) THEN
          HROXID(IHV+NHV) = ZROX0 * FPARH
          HROXID(3*IHV+NHV) = ZROX0 * FPARH
        ENDIF
C
      ENDIF
C
C -- CHECK INPUT DATA
C
      IF(NKSL.GE.-NTHIST .AND.NKSL.LE.IKS) GO TO 20
      NTHISM=-NTHIST
      WRITE(IOUT,1190 ) NKSL,NTHISM,IKS
20  IF(NKSR.GE.-NTHIST.AND.NKSR.LE.IKS) GO TO 30
      NTHISM=-NTHIST
      WRITE(IOUT,1200 ) NKSR,NTHISM,IKS
      CALL ERROR('HCINP ',16,1,'STOP IN HCINP',*995)
995  CONTINUE
C
C -- DETERMINE IF LEFT OR RIGHT SIDE ADIABATIC
C
30  LADR = 0
      IF (NKSL .EQ. 0) LADR=1
      IF (NKSR .EQ. 0) LADR=LADR + 2
C
      IF(LADR.NE.3) GO TO 40
      WRITE(IOUT,1210 ) NHV
      CALL ERROR('HCINP ',16,1,'STOP IN HCINP',*994)
994  CONTINUE

```



```

C
40 IF(NGEOR.GE.0.AND.NGEOR.LE.4) GO TO 50
    WRITE(IOUT,1220 ) NGEOR
    CALL SERROR('HCINP      ',16,1,'STOP IN HCINP',*993)
993 CONTINUE
C
50 IF(N2.GE.0.AND.N2.LE.NLAYM1) GO TO 60
    WRITE(IOUT,1230 ) N2,NLAYM1
    CALL SERROR('HCINP      ',16,1,'STOP IN HCINP',*992)
992 CONTINUE
C
60 I= NLAYM1 - 1
    IF(N2.GE.0.AND.N3.EQ.0) GO TO 70
    IF(N2.GT.0.AND.N3.GE.0.AND.N3.LE.I) GO TO 70
    WRITE(IOUT,1240 ) N3,N2,I
    CALL SERROR('HCINP      ',16,1,'STOP IN HCINP',*991)
991 CONTINUE
C
70 IF(NKSL.LE.0) GOTO 75
    IF(IMLK(NKSL).LT.0) GOTO 75
    IF (IFLOW(NKSL) .EQ. -3) THEN
        HCU(NHV) = ZERO
        HCO(NHV) = DHYI(NKSL)
        GOTO 75
    ENDIF
    IF(SLER8(HCO(NHV),HCU(NHV))) GOTO 71
    IF(SLER8(HCO(NHV),ZTI(NKSL)) .AND. SGER8(HCU(NHV),ZBI(NKSL)))
* GO TO 75
71 WRITE(IOUT,1245) NHV,NKSL
    WRITE(IOUT,*)HCO(NHV),HCU(NHV),ZTI(NKSL),ZBI(NKSL)
    CALL SERROR('HCINP      ',04,1,'          ',*990)
990 CONTINUE
75 IF(NKSR.LE.0) GOTO 76
    IF(IMLK(NKSR).LT.0) GOTO 76
    IF (IFLOW(NKSR) .EQ. -3) THEN
        HCU(NHV) = ZERO
        HCO(NHV) = DHYI(NKSR)
        GOTO 76
    ENDIF
    IF(SLER8(HCO(NHV),HCU(NHV))) GOTO 72
    IF(SLER8(HCO(NHV),ZTI(NKSR)) .AND. SGER8(HCU(NHV),ZBI(NKSR)))
* GO TO 76
72 WRITE(IOUT,1245) NHV,NKSR
    WRITE(IOUT,*)HCO(NHV),HCU(NHV),ZTI(NKSR),ZBI(NKSR)
    CALL SERROR('HCINP      ',04,1,'          ',*989)
989 CONTINUE
C
C
C -- OBTAIN TEMPERATURES FROM FLUID DYNAMICS
76 IF (NKSL .GT. 0 ) TTL(NHV) = TFHT(NKSL)
    IF (NKSR .GT. 0 ) TTR(NHV) = TFHT(NKSR)
C
C -- ADIABATIC USE TEMPERATURE FROM OTHER SIDE
C (ONLY FOR INITIALIZATION OF LAYER TEMPERATURES; WILL BE RESET AT
C THE END OF HCINP)
    IF (NKSL .EQ. 0 ) TTL(NHV) = TFHT(NKSR)
    IF (NKSR .EQ. 0 ) TTR(NHV) = TFHT(NKSL)
C
C -- IF TEMPERATURE IS INPUT AS F(TIME) USE TEMP FROM OTHER SIDE
C (ONLY FOR INITIALIZATION OF LAYER TEMPERATURES; WILL BE RESET AT
C THE END OF HCINP)
    IF (NKSL .LT. 0 ) TTL(NHV) = TFHT(NKSR)
    IF (NKSR .LT. 0 ) TTR(NHV) = TFHT(NKSL)
C
C -- CHECK TEMPERATURE FOR RANGE (DEFENSIVE PROGRAMMING)
    IF (TTL(NHV) .LT. 0 .OR. TTL(NHV) .GT. 1500)
* WRITE(IOUT,1320) NHV, TTL(NHV)
C
    IF (TTR(NHV) .LT. 0 .OR. TTR(NHV) .GT. 1500)
* WRITE(IOUT,1330) NHV, TTR(NHV)
C
LAY1=LAY1SV

```

```

C
C      DO 80   I=1,6
C          IF(PARTIT(I).LE.ZERO)  PARTIT(I)=ZERO
C
80  CONTINUE
C          IF(AORXL.LE.ZERO)  AORXL=ONE
C
      DO 90   I=1,3
      IF(CPL (I).LT.ZERO)  CPL (I)=HIGH
      IF(HTCL(I).LT.ZERO)  HTCL(I)=HIGH
      IF(RHOL(I).LT.ZERO)  RHOL(I)=HIGH
      IF(WLFL(I).LT.ZERO)  WLFL(I)=HIGH
90  CONTINUE
C
      IF(HTCL(4).LT.ZERO)  HTCL(4)=HIGH
      AREA   = AORXL
      PIXL   = PII*AORXL
      PIXL2  = PIXL/HALF
      NHV1   = NHV-1
C
      IF(NHV.LE.1)  GO TO 110
C
C  --  SUM UP TO CURRENT STARTING ADDRESS
C
      DO 100  I=1,NHV1
C
      N      = NOLAYS(I)
      LAY1SV = LAY1SV+N
      LAY2SV = LAY2SV+N+1
C
100  CONTINUE
C
110  N1      = NLAY - N2 - N3
C
C  --  STORE INDEX OF PROPERTIES INTO MATL(NN)
      IJK    = IMATL + N1 - 1
      IJK1   = IMATL
C      IJK1 WIRD NUR FUER DEN TESTAUSDRUCK 'MATL' GERBRAUCHT
C
C  --  STORE INDEX OF MATL PROPERTIES FOR LAYERS
      DO 120  NN = IMATL,IJK
      MATL(NN) = MATL1
120  CONTINUE
C
      IMATL   = IMATL + N1
      IF(N2.LE.0) GOTO 140
      IJK     = IMATL + N2 - 1
C
      DO 130  NN=IMATL,IJK
      MATL(NN) = MATL2
130  CONTINUE
C
      IMATL   = IMATL + N2
140  CONTINUE
      IF(N3.LE.0) GOTO 160
      IJK     = IMATL + N3 - 1
C
      DO 150  NN = IMATL,IJK
      MATL(NN) = MATL3
150  CONTINUE
C
      IMATL   = IMATL + N3
160  CONTINUE
C
      TESTPRINT FOR MATL
      IF(IOPTHC.GE.10)
*          WRITE(IOUT,1100 ) IJK1,IJK,NHV,N1,N2,N3,IMATL,
*          (MATL(NN),NN=IJK1,IJK)
C

```

```

C
C -- STORE STARTING INDEXES OF 1ST LAYERS OF MAT'LS IN EACH HTV
C   TO FACILITATE ACCESS LATER
NL1   = IHV + NHV
NL2   = IHV + NL1
NL3   = IHV + NL2
NL4   = IHV + NL3
NL5   = IHV + NL4
NL6   = IHV + NL5
C -- ADDRESS OF 1ST LAYER IN CURRENT HTV
NOLAYS(NL1) = LAY1SV
C -- ADDRESS OF 1ST LAYER OF 2ND MAT'L
NOLAYS(NL2) = LAY1SV + N1
C -- ADDRESS OF 1ST LAYER OF 3RD MAT'L
NOLAYS(NL3) = LAY1SV + N1 + N2
C -- ADDRESS OF 1ST SURFACE IN CURRENT HTV
NOLAYS(NL4) = LAY2SV
C -- ADDRESS OF 1ST SURFACE OF 2ND MAT'L
NOLAYS(NL5) = LAY2SV + N1
C -- ADDRESS OF 1ST SURFACE OF 3RD MAT'L
NOLAYS(NL6) = LAY2SV + N1 + N2
C
L0     = LAY2SV
LN1    = L0 + N1
C
C -- SUM UP WIDTHS TO RADIA
C
DO 170 I = 1,5
PARTIT(I+1) = DABS(PARTIT(I))+DABS(PARTIT(I+1))
170 CONTINUE
C
IF(NGEOR.GE.3) GO TO 240
C
C
C FIRST MATERIAL (LEFT)
C
R0     = PARTIT(1)
RN1    = PARTIT(2)
RV(L0) = R0
RV(LN1) = RN1
C
IF(NGEOR.EQ.2) GO TO 180
C
C -- PLATE OR CYLINDER WITH EQUAL THICKNESSES OF LAYERS: NGEOR=0,1
C   CYLINDER WITH CONSTANT VOLUME LAYERS : NGEOR=2
C
STEPE(1) = (RN1-R0)/FLOAT(N1)
GO TO 190
C
180 AUX   = R0 * R0
STEPV1(1) = AUX
STEPV2(1) = (RN1*RN1-AUX)/FLOAT(N1)
C
190 IF(N2.EQ.0) GO TO 230
C
C SECOND MATERIAL (MIDDLE)
C
LN2     = LN1 + N2
RN2     = PARTIT(4)
RV(LN2) = RN2
C
IF(NGEOR.EQ.2) GO TO 200
C
STEPE(2) = (RN2-PARTIT(3))/FLOAT(N2)
GO TO 210
C
200 AUX   = PARTIT(3) * PARTIT(3)
STEPV1(2) = AUX
STEPV2(2) = (RN2*RN2-AUX)/FLOAT(N2)
C
210 IF(N3.EQ.0) GO TO 230
C

```

```

C      THIRD MATERIAL (RIGHT)
C
C      LN3      = LN2 + N3
C      RN3      = PARTIT(6)
C      RV(LN3) = RN3
C
C      IF(NGEOR.EQ.2) GO TO 220
C
C      STEPE(3) = (RN3-PARTIT(5))/FLOAT(N3)
C      GO TO 230
C
C      220 AUX      = PARTIT(5)*PARTIT(5)
C      STEPVL(3) = AUX
C      STEPVL(3) = (RN3*RN3-AUX)/FLOAT(N3)
C
C      230 IF(NGEOR.NE.0.AND.NGEOR.NE.3.AND.RV(L0).EQ.ZERO) LADR=0
C      230 CONTINUE
C
C      -- DETERMINE RV,SDE,SV FOR INNER LAYERS OF ALL THREE MATERIALS
C
C      CALL HCEQVL(RV(LAY2SV),SDE,SV(LAY1SV),PARTIT,STEPE,STEPVL,STEPV2,
C      *          MATLAY,AREA,PIXL,NGEOR,NLAY,PLV(LAY1SV))
C
C      GO TO 250
C
C      -- FREE PATTERN FOR LAYERS OF PLATE OR CYLINDER: NGEOR=3,4
C
C      =====
C      RV FELDER MUESSEN NOCH BESETZT WERDEN.*
C      =====
C
C      240 CALL HCFREE(RV(LAY2SV),SDE,SV(LAY1SV),PARTIT,MATLAY,AREA,PIXL,
C      *          NGEOR,NLAY,PLV(LAY1SV),NERFRE)
C
C      DISTINGUISH ONLY BETWEEN PLATE AND CYLINDER FROM NOW ON
C
C      250 IF(NGEOR.EQ.3) NGEOR=0
C      IF(NGEOR.NE.0) NGEOR=1
C      IF(NGEOR.EQ.1.AND.RV(L0).EQ.ZERO) NGEOR=2
C
C      -- PLATE=0 ; HAWL CYLINDER=1 ; FULL CYLINDER IN THE CENTER=2
C      =====
C
C      -- INITIATE TEMPERATURE DISTRIBUTION WITHIN LAYERS
C
C      CALL HCTINI(RV(LAY2SV),SDE,TT(LAY1SV),TTL(NHV),TTR(NHV),LADR,
C      *          NGEOR,NLAY)
C
C      -- CALCULATE HEAT RESISTANCE FACTORS FOR MORE CONVENIENCE
C
C      CALL HCFSET(FWR1(LAY2SV),FWR2(LAY2SV),FWR3(LAY2SV),RV(LAY2SV),
C      *          SDE,PARTIT,MATLAY,AREA,PIXL2,LADR,NGEOR,NLAY)
C
C      AUX=PIXL2
C      IF(NGEOR.EQ.0) AUX=AREA
C      PCYLL (NHV)=AUX
C      IVOLL (NHV)=NKSL
C      IVOLR (NHV)=NKSR
C      LAD (NHV)=LADR
C      NGEOR (NHV)=NGEOR
C
C
C      -- IF ADIABATIC SET IALPHH TO BE CERTAIN
C      IF(NKSL .EQ. 0) IALPHH(1) = 0
C      IF(NKSR .EQ. 0 ) IALPHH(4) = 0
C      IF(NKSR .EQ. 0 .AND. N3 .EQ. 0) IALPHH(3) = 0
C      IF(NKSR .EQ. 0 .AND. N2 .EQ. 0) IALPHH(2) = 0
C
C
C      -- IF ZONE 2 OR ZONE 3 IS NOT DEFINED SET IALPHH TO BE CERTAIN
C      IF (N2 .EQ. 0) IALPHH(3) = 0
C      IF (N3 .EQ. 0) IALPHH(4) = 0
C
C
C      -- STORE MODE FOR CALCULATION OF HTC

```

```

      IALPH(LAY2SV)      = IALPHH(1)
      IALPH(LAY2SV + N1) = IALPHH(2)
      IF (N2 .GT. 0) IALPH(LAY2SV + N1 + N2) = IALPHH(3)
      IF (N3 .GT. 0) IALPH(LAY2SV + N1 + N2 + N3) = IALPHH(4)
C
C -- INITIALIZE HTC IF HTC IS TO BE CALCULATED
      IF (IALPHH(1) .EQ. 1 .AND. HTCL(1) .LE. ZERO) HTCL(1) = 3.0 D+03
      IF (IALPHH(2) .EQ. 1 .AND. HTCL(2) .LE. ZERO) HTCL(2) = 3.0 D+03
      IF (IALPHH(3) .EQ. 1 .AND. HTCL(3) .LE. ZERO) HTCL(3) = 3.0 D+03
      IF (IALPHH(4) .EQ. 1 .AND. HTCL(4) .LE. ZERO) HTCL(4) = 3.0 D+03
C
C -- CHECK MODE OF HTC CALCULATION FOR HCO COUPLED TO AN ACCUMULATOR
      IF (NKSL.GT.0) THEN
        IF (ICMPO(KL).EQ.1 .AND. IALPH(LAY2SV) .NE. 0)
          * CALL ERROR('HCINP ',8,1,ANAMH(KHCO))//
          * ': HCO IS CONNECTED TO AN ACCUMULATOR. THERE ARE NO HTC'
          * //' CORRELATIONS AVAILABLE FOR NITROGEN. USE OPTION "HTCL"'
          * //' IN THE "HTCDEF" INPUT SECTION.',*1000)
        ENDIF
        IF (NKSR.GT.0) THEN
          IF (ICMPO(KR).EQ.1 .AND. IALPH(LAY2SV+NLAY) .NE. 0)
            * CALL ERROR('HCINP ',8,1,ANAMH(KHCO))//
            * ': HCO IS CONNECTED TO AN ACCUMULATOR. THERE ARE NO HTC'
            * //' CORRELATIONS AVAILABLE FOR NITROGEN. USE OPTION "HTCL"'
            * //' IN THE "HTCDEF" INPUT SECTION.',*1000)
          ENDIF
        ENDIF
C
C -- INITIATE PROPERTIES OF MATERIALS WITH INPUT DATA
      CALL HCMSET(ATT(LAY1SV),
        * ATTFAC(LAY1SV),ATTL,CP(LAY1SV),CPL,HTC(LAY2SV),HTCL,
        * RHO(LAY1SV),RHOL,WLF(LAY1SV),WLFL,MATLAY,NGEOR,NLAY)
C
C
C -- CHECK FOR A S.S. HEAT TRANSFER
      IF (QTHRU(NHV) .EQ. ZERO) GO TO 280
C
C -- INITIALIZE HEAT FLOW TO/FROM FLUID VOLUME
      QHL(NHV) = -QTHRU(NHV)
      QHR(NHV) = QTHRU(NHV)
      IF (NKSL .GT. 0) QI(NKSL) = QI(NKSL) - QTHRU(NHV)
      IF (NKSR .GT. 0) QI(NKSR) = QI(NKSR) + QTHRU(NHV)
C
C
C
C
C 280 CONTINUE
C
      DO 290 I = 1,6
C
      IF(I.LT.6) PARTIT(7-I)=PARTIT(7-I)-PARTIT(6-I)
      IF((N2.EQ.0.AND.I.LE.4).OR.(N3.EQ.0.AND.I.LE.2)) PARTIT(7-I)=ZERO
C
C 290 CONTINUE
C
      RESET FLUID TEMPERATURES
      IF(NKSL.LE.0) TTL(NHV)=ZERO
      IF(NKSR.LE.0) TTR(NHV)=ZERO
C
C
      LAY1 = LAY1SV + NLAYM1
      LAY2 = LAY2SV + NLAY
C
C
C ISLAB=1
C
      K1 = 2 * MATL1 + 1
      K2 = 2 * MATL2 + 1
      K3 = 2 * MATL3 + 1
      IF(MATL1.LT.0) K1 = 1
      IF(MATL2.LT.0) K2 = 1
      IF(MATL3.LT.0) K3 = 1
C
      CONTROL NEW PAGE

```

```

NHVH1=NHV/2
NHVH2=(NHV+1)/2
IF(IOPTHC.LT.4) GO TO 410
IF(NHVH2.GT.NHVH1) WRITE(IOUT,1080 )
WRITE(IOUT,1041) NHV,IHTCS,ICHFS
C IF(IHTCS.EQ.1) WRITE(IOUT,1040) (IHTC(I),I=IFH,ISH)
C IF(ICHFS.EQ.1) WRITE(IOUT,1050) (ICHF(I),I=IFQ,ISQ)
WRITE(IOUT,1040) (IHTC(I),I=IFH,ISH)
WRITE(IOUT,1050) (ICHF(I),I=IFQ,ISQ)
AORL = XAREA
IF (NGEOR .GT. 0) AORL = XLENTH
NGEP1= NGEOR + 1
K1P1 = K1 + 1
K2P1 = K2 + 1
K3P1 = K3 + 1
WRITE(IOUT,1110 ) AORL, XMATL(K1), XMATL(K1P1), XMATL(K2),
* XMATL(K2P1),XMATL(K3),XMATL(K3P1),
* NHV,NKSL,TTL(NHV),NKSR,TTR(NHV),TYP(NGEP1),
* N1,N2,N3,(PARTIT(I),I=1,6),AORXL
WRITE(IOUT,1170 ) XNAME(1),(RV (I),I=LAY2SV,LAY2)
WRITE(IOUT,1170 ) XNAME(2),(SDE (I),I=1,NLAY)
WRITE(IOUT,1170 ) XNAME(3),(SV (I),I=LAY1SV,LAY1)
WRITE(IOUT,1170 ) XNAME(4),(ATTFAC(I),I=LAY1SV,LAY1)
WRITE(IOUT,1170 ) XNAME(5),(WLF (I),I=LAY1SV,LAY1)
WRITE(IOUT,1120 ) (IALPH (I),I=LAY2SV,LAY2)
WRITE(IOUT,1170 ) XNAME(6),(HTC (I),I=LAY2SV,LAY2)
WRITE(IOUT,1170 ) XNAME(7),(RHO (I),I=LAY1SV,LAY1)
WRITE(IOUT,1170 ) XNAME(8),(CP (I),I=LAY1SV,LAY1)
IF (IOPTHC .GT. 4)
*WRITE(IOUT,1170 ) XNAME(9),(TT (I),I=LAY1SV,LAY1)
WRITE(IOUT,1130 ) TDNBL(NHV),TDNBR(NHV),TFBL(NHV),TFBR(NHV),
* DHYL(NHV),DHYR(NHV)
WRITE(IOUT,1140 ) TWBAL(NHV),TWBAR(NHV),TWBEL(NHV),TWBER(NHV)
WRITE(IOUT,1150) QTHRU(NHV),TL(NHV),HLENGT(NHV),HCO(NHV),
* HCU(NHV)
C
410 CONTINUE
C
CDEI+ FUER ORTSLOTS
SOHH(IHVSUM+1) = 0.5D0 * HLENGT(1)
DO 415 I = 2,NHCV
SOHH(IHVSUM+I) = SOHH(IHVSUM+I-1) +
* 0.5D0 * (HLENGT(I-1) + HLENGT(I))
415 CONTINUE
CHDEI-
IHVSUM = IHVSUM + NHCV
C
C -- CREATE A PRINT PLOT OF HCO
C
IF(IPRPLO .EQ. 1 .AND. IOPTHC .GE. 3) THEN
CALL HPRPLO(HCO,HCU,HLENGT,KHCO,IA,IVOLL,IVOLR)
ENDIF
C
C -- FILL THE GRAPHIC FILE
C
IF(IGRPR .NE. 0) THEN
CALL HGRAPH(RV,HCU,HCO,KHCO,NHCOBJ,IA,IE,NOLAYS,IVOLL,IVOLR)
ENDIF
C
GO TO 1
C
999 CONTINUE
C
CDEI+ FO HECU PLOT HEADER
IAHO(KHCO+1) = IE + 1
NHGP = IHVSUM
NHCO = KHCO
CDEI-
IF(IOPTHC .GE. 1) THEN
WRITE(IOUT,'(//1X,130(1H-))')
ENDIF
C

```

```

      IF(LAYSUM .NE. NLAYMA .OR. IHVSUM .NE. IHV) THEN
        WRITE(IOUT, '( /1X,A,1I5/1X,A,1I5)') 'NLAYMA=', NLAYMA,
*       'NUMBER OF LAYERS READ IN:', LAYSUM
        WRITE(IOUT, '( /1X,A,1I5/1X,A,1I5/1X,A)') 'IHV=', IHV,
*       'NUMBER OF HCVS READ IN:', IHVSUM, 'STOP IN HCINP'
        CALL SERROR('HCINP      ',12,1,'STOP IN HCINP',*987)
987      CONTINUE
      ELSE
        WRITE(IOUT, '(' NUMBER OF HEATOBJECTS: ',1I6/)' )KHCO
C
        IF(IOPTHC .GE. 1) THEN
          WRITE(IOUT, '(1H1///30X,A/30X,A//)')
*         'S U M M A R Y', '=====
          WRITE(IOUT, '(2X,A/2X,79(1H-))')
*         ' #HCO HCO-NAME  NHV TFO-NAME(LEFT) TFO-NAME(RIGHT) MODEL'
*         '// '          #HCV FROM TO'
          J = 1
          DO 400 I = 1, KHCO
            IF(IVOLL(J) .LT. 0) TABKEL = STERN
            IF(IVOLL(J) .GE. 0) TABKEL = BLANK
            IF(IVOLR(J) .LT. 0) TABKER = STERN
            IF(IVOLR(J) .GE. 0) TABKER = BLANK
            IF(ICOMP0(I) .EQ. 0) MODE = MODEL(1)
            IF(ICOMP0(I) .EQ. 1) MODE = MODEL(2)
            IF(ICOMP0(I) .EQ. 2) MODE = MODEL(3)
            IF(I .EQ. 1) THEN
              ITABA = 1
              ITABE = NKHCO(I)
            ELSE
              ITABA = ITABA + NKHCO(I-1)
              ITABE = ITABA + NKHCO(I) - 1
            ENDIF
C+KONWAR-Erweiterung
            IF (ANAMH(I)(1:2) .EQ. 'WR') THEN
              DO 401 IT = ITABA, ITABE
                WRHCO(IT) = 1
                WRITE(IOUT,*) ' ANAMH(I) : ', ANAMH(I)
                WRITE(IOUT,*) ' WRHCO(IT) = ', WRHCO(IT)
401              CONTINUE
                ITABAI(I) = ITABA
                ITABEL(I) = ITABE
              ELSE
                DO 402 IT = ITABA, ITABE
                  WRHCO(IT) = 0
                  WRITE(IOUT,*) ' ANAMH(I) : ', ANAMH(I)
                  WRITE(IOUT,*) ' WRHCO(IT) = ', WRHCO(IT)
402              CONTINUE
            ENDIF
C-KONWAR-Erweiterung
            J = J + NKHCO(I)
            WRITE(IOUT, '(2X,1I5,2X,1A10,1X,1I2,4X,1A1,1A10,4X,1A1,1A10
*            ,2X,1A14,4X,1I3,1X,1I3,3X)')
*            I, ANAMH(I), NKHCO(I), TABKEL, AOLH(I), TABKER, AORH(I), MODE,
*            ITABA, ITABE
400          CONTINUE
        ENDIF
C+AUH
        WRITE(IOUT, '( /1X,1A1,A)')
*       STERN, ' : SIGNAL NAME (BOUNDARY CONDITION)'
C-AUH
        ENDIF
        NHOBJ = KHCO
C
        CALL SERROR('HCINP      ',4,-1,' ALL HCOS READ IN'//
*        ' AND ALL HCVS GENERATED.',*955)
955      CONTINUE
C
        IF (IIIMAT .NE. 0) THEN
          CALL SERROR('HCINP      ',16,1,'IIIMAT .NE. 0',*967)
967        CONTINUE
        ENDIF
C -- INPUT ERRORS FOR FREE GEOMETRY

```

```

      IF(NERFRE.GT.0) GOTO 380
C
C
      IMATL = IMATL - 1
      IF(LAYAL1.NE.IMATL) WRITE(IOUT,1090 ) LAYAL1,IMATL
C
      IF (IOPTHC .GE.5) CALL HCPRNT
      WRITE(IOUT,1080 )
C
C -- OBTAIN HEAT TRANSFER COEFFICIENT DEFINED AS A FUNCTION OF
C TIME
C+AUH
C      IF (NALPHA .EQ. 0 ) GOTO 320
C      IF (IOPTHC .GE.3) WRITE(IOUT,1260 )
C      CALL HCFUNC(IALPHA, NALPHA, MFUNCT, NPTS,
C      *           XARRAY, YARRAY, FNAME)
C-AUH
C
C -- OBTAIN TEMPERATURE DEPENDENT MATL PROP
C 320 CONTINUE
      IF (NMPROP .EQ. 0 ) GOTO 330
      IF (IOPTHC .GE.3) WRITE(IOUT,1270 )
      CALL HCFUNC(ICP, NMPROP, MFUNCT,NPTS,
      *           XARRAY,YARRAY,FNAME)
      IF (IOPTHC .GE.3) WRITE(IOUT,1280 )
      CALL HCFUNC(IWLF,NMPROP,MFUNCT,NPTS,
      *           XARRAY,YARRAY,FNAME)
      IF (IOPTHC .GE.3) WRITE(IOUT,1290 )
      CALL HCFUNC(IRHO,NMPROP ,MFUNCT,NPTS,
      *           XARRAY,YARRAY,FNAME)
      330 CONTINUE
C
C -- OBTAIN TEMPERATURE AS A FUNCTION OF TIME
C+AUH
C      IF (NTHIST .EQ. 0 ) GOTO 340
C      ITEM = IRHO + NMPROP
C      IF (IOPTHC .GE.3) WRITE(IOUT,1300 )
C      CALL HCFUNC(ITEM,NTHIST, MFUNCT,NPTS,
C      *           XARRAY,YARRAY,FNAME)
C-AUH
      340 CONTINUE
C
C -- READ IN OXIDATION TABLES
C
      IF(IOXTF .EQ. 1) THEN
C
      NTABZR = 0
      DO 600 IOB = 1 , NHOBJ
      IF(MODOXI(IOB) .EQ. 0) GO TO 600
      IF(AZRRHO(IOB) .EQ. 'DUMMY') GO TO 600
C
      IF(NTABZR .EQ. 0) THEN
      NTABZR = NTABZR + 1
      AZRNAM(NTABZR) = AZRRHO(IOB)
C
      IF(NTABZR .GT. MFUNOX) THEN
      CALL ERROR('HCINP ',16,1,'TOO MANY OXIDATION'//
      *           ' FUNCTIONS',*1000)
      ENDIF
C
      ELSE
C
      DO 610 IT = 1 , NTABZR
      IF(AZRNAM(IT) . EQ. AZRRHO(IOB)) GO TO 611
610 CONTINUE
C
      NTABZR = NTABZR + 1
      AZRNAM(NTABZR) = AZRRHO(IOB)
C
      IF(NTABZR .GT. MFUNOX) THEN
      CALL ERROR('HCINP ',16,1,'TOO MANY OXIDATION'//
      *           ' FUNCTIONS',*1000)

```



```

                ENDIF
C
611            CONTINUE
                ENDIF
C
600            CONTINUE
C
                NTABZO = 0
                DO 700 IOB = 1 , NHOBJ
                    IF(MODOXI(IOB) .EQ. 0) GO TO 700
                    IF(AZO2RH(IOB) .EQ. 'DUMMY') GO TO 700
C
                    IF(NTABZO .EQ. 0) THEN
                        NTABZO = NTABZO + 1
                        NTABTO = NTABZR + NTABZO
                        AZRNAM(NTABTO) = AZO2RH(IOB)
C
                        IF(NTABTO .GT. MFUNOX) THEN
                            CALL SERROR('HCINP      ',16,1,'TOO MANY OXIDATION'//
*                               ' FUNCTIONS',*1000)
                                ENDIF
C
                        ELSE
C
                            NTABTO = NTABZR + NTABZO
                            DO 710 IT = NTABZR+1 , NTABTO
                                IF(AZRNAM(IT) .EQ. AZO2RH(IOB)) GO TO 711
110            CONTINUE
C
                            NTABZO = NTABZO + 1
                            NTABTO = NTABZR + NTABZO
                            AZRNAM(NTABTO) = AZO2RH(IOB)
C
                            IF(NTABTO .GT. MFUNOX) THEN
                                CALL SERROR('HCINP      ',16,1,'TOO MANY OXIDATION'//
*                               ' FUNCTIONS',*1000)
                                    ENDIF
C
711            CONTINUE
                                ENDIF
C
700            CONTINUE
C
                NTABTO = NTABZR + NTABZO
C
                IF(NTABZR .GT. 0) THEN
                    CALL HCFUNC(1,NTABZR,NTABTO,NPZRXY,ZRVALX,ZRVALY,AZRNAM)
                ENDIF
C
                IF(NTABZO .GT. 0) THEN
                    NT = NTABZR + 1
                    CALL HCFUNC(NT,NTABZO,NTABTO,NPZRXY,ZRVALX,ZRVALY,AZRNAM)
                ENDIF
C
            ENDIF
C
            ENDIF
C
            VORBELEGUNG VON HTC-ALT
            DO 365 I=1,LAYAL2
365            HTCA(I)=HTC(I)
C
C
C
            FOR INITIATION OF HTC FOR M-L (IN HCSTA)
            IGL=-1
C
            IF(IOPTHC.GE.5) CALL HCPRNT
C
            IF(IOPTHC .GE. 3) THEN
                WRITE(IOUT, '(//1X,30X,A)') '===== '
                WRITE(IOUT, '(1X,30X,A)') '* END OF HCINP *'
                WRITE(IOUT, '(1X,30X,A//////////)') '===== '
                CPUEL = TANF - FLOAT(MTIME(0))
                WRITE(IOUT, '(//1X,A,1F10.2,A)')

```

```

*'ELAPSED CPU-TIME FOR HECU INITIATION: ',CPUEL,' SECONDS.'
ENDIF
C
C KONWAR-Erweiterung
NHV = 1
DO 901 I=1,IHV
  QUOTHC(NHV) = (HCO(NHV) - HCU(NHV))/HLENGT(NHV)
  IF (QUOTHC(NHV) .GT. 1.0D+00) QUOTHC(NHV) = 1.0D+00
  ANEIG(NHV) = 180./KPI*DASIN(QUOTHC(NHV))
  WRITE (IOUT,*) ' NHV = ', NHV, ' NEIGUNG = ', ANEIG(NHV)
  WRITE (66,*) ' NHV = ', NHV, ' NEIGUNG = ', ANEIG(NHV)
  WRITE (66,*) ' HCO = ', HCO(NHV), ' HCU = ', HCU(NHV),
  * ' HLENGT = ', HLENGT(NHV)
  WRITE(IOUT,*)
  * 'NUMBER OF HCVs :', NHCV
  WRITE(IOUT,*) ' Kontrollindex: ', WRHCO(I)
  NHV = NHV + 1
901 CONTINUE
C KONWAR-Erweiterung
1000 RETURN
C
380 WRITE(6,1340) NERFRE
CALL SERROR('HCINP ',16,1,'STOP IN HCINP',*986)
986 CONTINUE
C
1040 FORMAT(' IHTC(1...3)',/, ' ',3I8)
1041 FORMAT('0 IHTCS ICHFS ',30('*'),' HEAT SLAB NO.',I4,
*' ',65('*'),/, ' ',2I8)
1050 FORMAT(' ICHF IDNBFB IFB IRNB IRNBNB INB',/,
*' ',6I8)
1080 FORMAT(1H1)
1090 FORMAT(/5X,'LAYAL1 = ',I3,3X,'.NOT EQUAL. IMATL = ',I3/)
1100 FORMAT(//' MATL(' ',I3,') - MATL(' ',I3,')',10X,'NHV = ',I2,
*' N1 = ',I2,' N2 = ',I2,' N3 = ',I2,' IMATL = ',I2,
*/(10X,20I5)/)
1110 FORMAT('0SLAB LEFT LEFT RIGHT RIGHT GEO. LAYERS
2 INNER THICKNESSES OF
3',1A8,/,
4 ' # VOL. TEMP. VOL. TEMP. TYPE N1 N2 N3
5 RADIUS MATERIAL 1 GAP 1 MATERIAL 2 GAP 2 MATERIAL 3
* ',/,68X,2A8,6X,2A8,6X,2A8,
6 /,1X,I3,I6,F9.1,I5,F8.1,1X,A8,1X,4X,I4,2I3,1P,7D11.3,/)
1120 FORMAT(' ',6X,'IALPH (I) ',9I12,/, ' ',19X,9I12)
1130 FORMAT(/,7X,'TDNBL',7X,'TDNBR',7X,'TFBL',8X,'TFBR',8X,'DHYL',
* 8X,'DHYR',/, ' ',5X,1P,6D12.5)
1140 FORMAT(/,7X,'TWBAL ',5X,'TWBAR ',5X,'TWBEL ',5X,'TWBER ',/,
* 6X,1P,6D12.5)
1150 FORMAT(/,7X,'QTHRU',7X,'TL',10X,'HLENGT',6X,'HCO',9X,'HCU',
*/, ' ',5X,1P,6D12.5)
1170 FORMAT(1H ,6X,A6,'(I) ',1P,9D12.5,/, (19X,9D12.5))
1190 FORMAT('0+++++ HECU +++++ INPUT WARNING : LEFT VOLUME # IS ',I4,
*' , SHOULD BE BETWEEN',I4,' AND',I4,/, ' ',34X,'PROGRAM STOP IN HC
*INP')
1200 FORMAT('0+++++ HECU +++++ INPUT WARNING : RIGHT VOLUME # IS ',I4,
*' , SHOULD BE BETWEEN',I4,' AND',I4,/, ' ',34X,'PROGRAM STOP IN HC
*INP')
1210 FORMAT('0+++++ HECU +++++ INPUT WARNING : BOTH SIDES ADIABATIC',
*/, ' ',34X,'CHECK NKSL AND NKSR',/, ' ',34X,'PROGRAM STOP IN HCINP')
1220 FORMAT('0+++++ HECU +++++ INPUT ERROR : GEOMETRY KEY IS ',I4,
*' , SHOULD BE BETWEEN 0 AND 4',/, ' ',34X,'PROGRAM STOP IN HCINP')
1230 FORMAT('0+++++ HECU +++++ INPUT WARNING : # OF LAYERS IN SECOND M
*ATERIAL IS ',I2,' , SHOULD BE BETWEEN 0 AND ',I2,/, ' ',34X,
*'PROGRAM STOP IN HCINP')
1240 FORMAT('0+++++ HECU +++++ INPUT WARNING : # OF LAYERS IN THIRD M
*ATERIAL IS ',I2,' , SHOULD BE BETWEEN 0 AND ',I2,/,35X,' OR SECO
*ND MATERIAL NON-EXISTENT',/, ' ',34X,'PROGRAM STOP IN HCINP')
1245 FORMAT('0+++++ HECU +++++ INPUT WARNING : GEOMETRIC HEIGHT OF CON
*DUCTOR NHV = ',I3,' DOES NOT COINCIDE WITH',/, ' GE
*OMETRIC HEIGHT OF CONTROL VOLUME I = ',I3,/, ' OR H
*CO(NHV) .LE. HCU(NHV)',/, ' CHECK ',
*'HCO(NHV), HCU(NHV), ZTI(I) AND ZBI(I)')
C1260 FORMAT('0HEAT TRANSFER COEFFICIENT AS A FUNCTION OF TIME')

```

```
1270 FORMAT('HEAT CAPACITY AS A FUNCTION OF TEMPERATURE')
1280 FORMAT('THERMAL CONDUCTIVITY AS A FUNCTION OF TEMPERATURE')
1290 FORMAT('DENSITY AS A FUNCTION OF TEMPERATURE')
C1300 FORMAT('TEMPERATURE AS A FUNCTION OF TIME')
1320 FORMAT(///,'***** WARNING TTL(' ,I4,' ) = ' ,1P,D14.5,
*      ' ***** --HECU--',//)
1330 FORMAT(///,'***** WARNING TTR(' ,I4,' ) = ' ,1P,D14.5,
*      ' ***** --HECU--',//)
1340 FORMAT(///,'*****',I4,' INPUT ERROR DETECTED FOR FREE GEOMETRY',
*      /,' ***** INPUT ERRORS DETECTED IN HCFREE',
*      /,' ***** PROGRAMM STOP IN HCINP')
END
```



# Anhang E Unterprogramme MHTCN1 und MHTCN1A

## MHTCN1

```

SUBROUTINE MHTCN1 (TF,P,SQH,SQM,SQV,V,VW,VD,HW,HWD,QH,GF,GFD,GFW,
*               TTS,TTW,TSAE,HTC,HTCK,DHY,HLENGT,POL,MODE,NHV,
*               IHTCI,LWRITE,VST,HTCV,IEQUH)
CH+
CN      MHTCN1
CA      POI
CM      LER 21. 2.94: CARPENTER+COLBURN: REYNOLDS-LIMITATION IMPROVED
C*
CV      MOD1.1A
C*
CP      DETERMINATION OF HEAT TRANSFER COEFFICIENTS
CP      FOR HEAT FLOW FROM FLUID TO WALL: TFLUID > TWALL
C*
CI      DHY          : HYDRAULIC DIAMETER OF COOLANT CHANNEL          M
CI      HLENGT       : LENGTH OF HTV
CI      HTCK         : OLD HEAT TRANSFER COEF. (E.G. AT BEGIN OF TIME STEP)
CI                   : W/M*M*K
CI      HW           : SPECIFIC ENTHALPY OF SATURATED WATER          J/KG
CI      HWD          : HEAT OF EVAPORATION                          J/KG
CI      IEQUH        : INDICATOR IF SPLITTING OF HEAT FLUX IS DONE
CI                   : FOR 2E-CV'S
CI      IHTCI ( )    : INDEX OF CORRELATION TO BE USED FOR DETERMINATION
CI                   : OF HTC
CI      LWRITE       : CONTROLS OUTPUT
CI      NHV          : INDEX OF HTV
CI      POL ( )      : INTERPOLATION BOUNDARYS FOR A COSINE FORM
CI                   : TRANSITION BETWEEN HTC CORRELATIONS
CI      QH           : HEAT FLUX AT WALL                            W/M*M
CI      SQH          : ENTHALPY VOID FRACTION  $X=H-H'/R$               -
CI      SQV          : VOID FRACTION                                -
CI      TF           : PROBLEM TIME                                  S
CI      TSAE         : SATURATION TEMPERATURE                      DEG C
CI      TTS          : SURFACE TEMPERATURE                         DEG C
CI      TTW          : COOLANT TEMPERATURE                         DEG C
CI      V            : SPECIFIC VOLUME OF COOLANT                  M*M*M/KG
CI      VD           : SPECIFIC VOLUME OF SATURATED STEAM          M*M*M/KG
CI      VW           : SPECIFIC VOLUME OF SATURATED WATER          M*M*M/KG
CI      SQM          : MASS FLOW QUALITY
C*
CU      GF           : MASS FLUX                                    KG/S*M*M
CU      GFD          : VAPOUR MASS FLUX                            KG/S*M*M
CU      GFW          : WATER MASS FLUX                             KG/S*M*M
CU      HTC          : HEAT TRANSFER COEFF. TO BE CALCULATED       W/M*M*K
CU      MODE         : INDEX OF CORRELATION USED FOR DETERMINATION OF HTC -
CU      P            : COOLANT PRESSURE                             PA
CU      VST          : ABSOLUTE VALUE OF REL. VELOCITY
CU      HTCV         : HEAT TRANSFER COEF. FOR WALL - VAPOR HEAT TRANSFER
CU                   : W/M*M*K
C*
CF      MPTAF
C*
CR      ERRTRA      MPSPLK  MPZUS1  SINPO
C*
CB      MHTCN      MHTCQF
C*
CH-

```

```

      INCLUDE (C)
      INCLUDE (CA)
      INCLUDE (CDSS16)
      INCLUDE (CSEBPI)
      INCLUDE (ANDREA)
      INCLUDE (CHCDN4)

C      DIMENSION POL      (000014),IHTCI (000003)

CIBM+
      DOUBLE PRECISION MPTAF

CIBM-
CCVX+
C      DOUBLE PRECISION MPTAF_P101,MPTAF_P104
CCVX-
      LOGICAL LWRITE, LWRT1

C
      PARAMETER (ZERO = 0.D0)
      PARAMETER (ONE  = 1.D0)
      PARAMETER (IOUT = 6 )
      PARAMETER (RETURB = 5.D3)

CMIS+
C      DEFAULT VALUE FOR HTCv
C
C      OUTPUTAKTIVIERUNG alle 25 Sekunden
C      IF (TF-TFA .GE.25.0) THEN
C          WRITE(66,*) 'MHTCN1as.F Problemzeit = ', TF
C          LWRT1=.TRUE.
C          TFA=TF
C          IZAEHL=1
C      ELSE
C          IZAEHL=IZAEHL+1
C          IF (IZAEHL.LE.4) THEN
C              LWRT1=.TRUE.
C          ELSE
C              LWRT1=.FALSE.
C          ENDIF
C      ENDIF
C      IF (TF-TFA .EQ. 50.0 .AND. NHV .EQ. 1) THEN
C          WRITE(66,*) 'AUFRUF MHTCN1 FUER NHV = ', NHV, ' AT ', TF
C          DO 1 NV = 1,100
C              WRITE(66,*) ' NHV = ', NV, ' WRHCO(NHV) = ', WRHCO(NV)
C          CONTINUE
C      ENDIF
C      IF (LWRT1 .AND. NHV .EQ. 1) THEN
C          WRITE(66,*) 'AUFRUF MHTCN1 FUER NHV = ', NHV, ' AT ', TF
C          DO 2 NV = 1,100
C              WRITE(66,*) ' NHV = ', NV, ' WRHCO(NHV) = ', WRHCO(NV)
C          CONTINUE
C      ENDIF
C      IF (LWRT1) WRITE (66,*) 'AUFRUF MHTCN1 FUER NHV = ', NHV,
C          *      ' mit WRHCO = ', WRHCO(NHV)
C      IF (WRHCO(NHV) .EQ. 1) THEN
C          IF (LWRT1) WRITE (66, *) 'AUFRUF MHTCN1as FUER ', ANAMH(NHV)
C          CALL MHTCN1A(TF,P,SQH,SQM,SQV,V,VW,VD,HW,HWD,QH,GF,GFD,GFW,
C          *      TTS,TTW,TSAE,HTC,HTCK,DHY,HLENGT,POL,MODE,NHV,
C          *      IHTCI,LWRITE,VST,HTCV,IEQUH)
C      ELSE
C          IF (LWRT1) WRITE (66, *) 'AUFRUF MHTCN1 FUER ', ANAMH(NHV)
C          HTCV=ZERO
CMIS-
C
C      CHECK INPUT DATA
C      IF (HTCK.LE.ZERO.OR.DHY.LE.ZERO.OR.HLENGT.LE.ZERO) GOTO 250
C      IF (.NOT.LWRITE) GOTO 10
C      WRITE(IOUT,1030)
C      WRITE(IOUT,*) NHV,TF,P,SQH,V,VW,VD,HW,HWD,QH,GF,TTW,TSAE,
C      *HTC,HTCK,DHY,HLENGT,(POL(IW),IW=1,4),MODE,
C      *(IHTCI(IW),IW=1,3),SQV,GFW,GFD
C
CSTF 4.12.81
10 CONTINUE
      IF (GF.LT.1.D-10) GF=1.D-10

```

```

      IF(GFW.LT.1.D-10) GFW=1.D-10
      IF(GFD.LT.1.D-10) GFD=1.D-10
C
      HTCMAC=ZERO
      HTCC=ZERO
C
      CALL MPSPLK(P,DP,DQ)
C
C
      IF (SQV.GE.POL(4)) GOTO 150
C
C-----
C*****SUBCOOLED FORCED CONVECTION*****
C-----
C-----DITTUS-BOELTER CORRELATION I-----
C-----MODE=1-----
C
      PH1=P
      HH1=(TTS+TTW)*5.D-1
CIBM+
      CALL MPZUS1(DP,DQ,PH1,HH1,3,1,ETAW,1,CEPW,1,WLFW)
CIBM-
CCVX+
C      CALL MPZUS1_3A(DP,DQ,PH1,HH1,ETAW,CEPW,WLFW)
CCVX-
      ETA1=ETAW
      CEP1=CEPW
      WLF1=WLFW
      IF (SQH.GE.POL(2)) GOTO 60
      SQHM=SQH
      IF(SQH.GT.POL(1)) SQHM=POL(1)
      RE=DHY*GFW/ETA1
C      WRITE(6,*) ETA1,CEP1,WLF1,ETA2,RE
      H19=SQHM
      IF(SQHM.LT.1.D-6) SQHM=1.D-6
      IF(SQHM.GT.9.9999D-1) SQHM=9.9999D-1
      REW=(ONE-SQHM)*RE
      RE8=REW**0.8D0
      PR=CEP1*ETA1/WLF1
      PR4=PR**0.4D0
      HTC5=0.023D0*RE8*PR4*(WLF1/DHY)
C      WRITE(6,*) HTC5,RE8,PR4,WLF1,DHY,FCH
      SQHM=H19
      MODE=1
C
C      60 CONTINUE
C
C-----NATURAL CONVECTION -----
      TFILM=(TTW+TTS)*5.D-1
      IF(TFILM.GT.TSAE) TFILM=TSAE
CIBM+
      CALL MPZUS1(DP,DQ,PH1,TFILM,3,1,ETAW,1,CEPW,1,WLFW)
CIBM-
CCVX+
C      CALL MPZUS1_3A(DP,DQ,PH1,TFILM,ETAW,CEPW,WLFW)
CCVX-
      PR=CEPW*ETAW/WLFW
CIBM+
      RHOWF=1.D0/MPTAF(101,PH1,TFILM)
      BETA=MPTAF(104,PH1,TFILM)
CIBM-
CCVX+
C      RHOWF=1.D0/MPTAF_P101(PH1,TFILM)
C      BETA=MPTAF_P104(PH1,TFILM)
CCVX-
      GRAS=RHOWF*RHOWF* EB *DABS((TTS-TTW)*BETA)/(ETAW*ETAW)
C      WRITE(6,*) NHV,RHOWF,ETAW,BETA,TTS,TTW
      HH1=GRAS*PR
      HH1=HH1**0.3333D0
      EXP1=-9.D0/1.6D1
      EXP2=-16.D0/27.D0

```

```

FPR=(1.D0+(2.D0*PR)**EXP1)**EXP2
HTCL=0.15D0*WLFW*HH1*FPR
IF(HTCL.LT.2.0D1) HTCL=2.0D1
IF(SQH.GE.POL(2)) GOTO 61
IF(HTCL.GT.HTC5) HTC5=HTCL
IF(SQH.GT.POL(1)) GOTO 61
HTC=HTC5
CMIS+
C   HTC>0 FOR SQM>0 (NOT DEPENDENT ON POL(1,2) )
   IF (SQM .GT. ZERO .AND. IEQUH .GT. 1) GOTO 150
CMIS-
GO TO 240
C-----
C---SUBCOOLED AND SATURATED NUCLEATE BOILING-----
C-----
C---BERECHNUNG VON HCMAC NACH CHEN-----
C-----
C-----MODE=5-----
C
C LER IF CONDENSATION ON WALL: NO MICROSCOPIC HEAT TRANSFER
61 SQHM=SQH
   IF(SQH.LT.POL(2)) SQHM=POL(2)
   SQHM3=POL(3)/VD/((1.D0-POL(3))/VW+POL(3)/VD)
   IF(SQH.GT.SQHM3) SQHM=SQHM3
   PH1=P
   HH1=(TTS+TTW)*5.D-1
   IF(SQH.GE.ZERO) GO TO 70
CIBM+
CALL MPZUS1(DP,DQ,PH1,HH1,3,1,ETAW,1,CEPW,1,WLFW)
CIBM-
CCVX+
C   CALL MPZUS1_3A(DP,DQ,PH1,HH1,ETAW,CEPW,WLFW)
CCVX-
ETA1=ETAW
CEP1=CEPW
WLF1=WLFW
GO TO 80
CIBM+
70 CALL MPZUS1(DP,DQ,PH1,HH1,1,1,STOF1,1,STOF2,1,STOF3)
CIBM-
CCVX+
C 70 CALL MPZUS1_1A(DP,DQ,PH1,HH1,STOF1,STOF2,STOF3)
CCVX-
ETA1=STOF1
CEP1=STOF2
WLF1=STOF3
C
CIBM+
80 CALL MPZUS1(DP,DQ,P,HH1,2,1,STOF1,1,STOF2,1,STOF3)
CIBM-
CCVX+
C 80 CALL MPZUS1_2A(DP,DQ,P,HH1,STOF1,STOF2,STOF3)
CCVX-
ETA2=STOF1
CEP2=STOF2
WLF2=STOF3
RE=DHY*GF/ETA1
C   WRITE(6,*) ETA1,CEP1,WLF1,RE
C   !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
   IF(RE.LT.1.0D+2) RE=1.0D+2
C   !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
H19=SQHM
IF(SQHM.LT.1.D-6) SQHM=1.D-6
IF(SQHM.GT.9.9999D-1) SQHM=9.9999D-1
REW=(ONE-SQHM)*RE
RE8=REW**0.8D0
PR=CEP1*ETA1/WLF1
PR4=PR**0.4D0
HH1=(ETA2/ETA1)**0.1D0
HH2=(VD/VW)**0.5D0
HH3=(SQHM/(ONE-SQHM))**0.9D0
HH1=HH1*HH2*HH3
HH1=HH1**0.8174D0

```



```

      FCH=ONE+1.6D0*(HH1)
      HTCMAc=0.023D0*RE8*PR4*(WLF1/DHY)*FCH
      IF(HTCMAc.LT.HTCL) HTCMAc=HTCL
      SQHM=H19
C-----
C-----BERECHNUNG DER KONDENSATION NACH NUSSELT (MOD. NACH MINCOWYCZ UND -
C-----ROSENOW-----
C-----MODE=5-----
C
      SQVM=SQV
C      WRITE(6,*) SQVM,POL(3),TTS,TSAE,TTW
      IF(SQVM.GT.POL(3)) SQVM=POL(3)
      IF(TTS.GT.TSAE) GOTO 85
      TTV=TTW
      IF(TTW.GT.TSAE) TTV=TSAE
      HH1=TTS+3.1D-1*(TTV-TTS)
CIBM+
      CALL MPZUS1(DP,DQ,P,HH1,3,1,STOF1,1,STOF2,1,STOF3)
CIBM-
CCVX+
C      CALL MPZUS1_3A(DP,DQ,P,HH1,STOF1,STOF2,STOF3)
CCVX-
      ETA1=STOF1
      CEP1=STOF2
      WLF1=STOF3
C      WRITE(6,*) ETA1,WLF1,TTW
      HH1=ETA1*HLENGT*WLF1*(TTV-TTS)
      HH2=1.D0/VW
      HH3=1.D0/VD
      HWDX=HWD*(1.D0+6.8D-1*CEP1*(TTV-TTS)/HWD)
      HH4=EB*HWDX*HH2*(HH2-HH3)
      HH5=(HH1/HH4)**2.5D-1
C      WRITE(6,*) HH1,HH4,HH5
      DELTA1=1.0606D0*HH5
      DELTA2=5.D-1*DHY*(1.D0-DSQRT(SQVM))
      DELTA=DMAX1(DELTA1,DELTA2)
      DELTA=DELTA/DMAX1(SQVM**1.D-1,1.D-5)
      IF(DELTA.LT.1.D-10) DELTA=1.D-10
      HTCC1=WLF1/DELTA
C-----NUSSELT-----
      RE=DHY*GFW/ETA1
      IF(RE.LT.1.D1) RE=1.D1
      IF(RE.GT.2.3D+3) RE=2.3D+3
      HH1=1.D0/VW
      HH2=1.D0/VD
      HH3=HH1*(HH1-HH2)*EB/ETA1/ETA1
      RE3=1.D0/RE**3.33D-1
      HH3=HH3**3.33D-1
      HTCC3=1.1025D0*RE3*HH3*WLF1
      HTCC4=DMIN1(HTCC1,HTCC3)
      HTCC5=HTCC4*(TTV-TTS)/(TTW-TTS)
      HTCC5=DMAX1(HTCMAc,HTCC5)
C-----CARPENTER UND COLBURN-----
      HH1=TTW
      IF(TTW.LT.TSAE) HH1=TSAE
CIBM+
      CALL MPZUS1(DP,DQ,P,HH1,4,1,STOF1,1,STOF2,1,STOF3)
CIBM-
CCVX+
C      CALL MPZUS1_4A(DP,DQ,P,HH1,STOF1,STOF2,STOF3)
CCVX-
      ETA2=STOF1
      CEP2=STOF2
      WLF2=STOF3
      REG=DHY*GFD/ETA2
C+LER
      REG25=DMAX1(RETURB,REG)**2.5D-1
C-LER
      IF(REG25.LT.1.D0) REG25=1.D0
      HH1=1.D0/VD
      HHG=GFD/HH1
      HH2=DMAX1(VST,HHG)
      HH2=HH2*HH2

```

```

      HH3=5.D-1*HH1*HH2
      THA=7.92D-2*HH3/REG25
      PR=CEP1*ETA1/WLF1
      PR5=PR**5.D-1
      THA5=THA**5.D-1
      HH4=1.D0/VW
      HH4=HH4**5.D-1
      HTCC6=6.5D-2*HH4*PR5*THA5*WLF1/ETA1
      HTCC6=HTCC6*(TTV-TTS)/(TTW-TTS)
      HTC=DMAX1(HTCC5,HTCC6)
C
      85 HTC1=HTC
      MODE=5
      IF(SQH.LT.POL(2)) GOTO 225
      IF (SQV.GT.POL(3))GO TO 150
CMIS+
      IF (IEQUH .GT. 1) GO TO 150
CMIS-
      GO TO 240
C
C-----
C*****SINGLE PHASE STEAM*****
C-----
C
C-----DITTUS-BOELTER-CORRELATION-II-----
C-----MODE=9-----
C
      150 PH1=P
      HH1=(TTW+TTS)*5.D-1
C
CSTF  4.12.81 KOPIE VON WAH
      IF (HH1.LT.TSAE) HH1=TSAE
CSTF  4.12.81 KOPIE VON WAH
C
CIBM+
      CALL MPZUS1(DP,DQ,PH1,HH1,4,1,ETAD,1,CEPD,1,WLFD)
CIBM-
CCVX+
C      CALL MPZUS1_4A(DP,DQ,PH1,HH1,ETAD,CEPD,WLFD)
CCVX-
      ETA=ETAD
      CEP=CEPD
      WLW=WLFD
      RE=DHY*GFD/ETA
      IF(RE.LT.5.D3) RE=5.D3
C      WRITE(6,*) ETA,CEP,WLW,RE
      HH2=RE**8.D-1
      PR=CEP*ETA/WLW
      HH3=PR**4.D-1
C
      IF(IHTCI(3).EQ.2) GOTO 160
      HH1=2.3D-2*WLW/DHY
CMIS+ HTC =HH1*HH2*HH3
      HTCD=HH1*HH2*HH3
CMIS- MODE=9
C      WRITE(6,*) HTC,HH1,HH2,HH3,PR
      GOTO 170
C
C-----MC ELIGOT-----
C-----IHTCI(3)=2-----MODE=9-----
C
      160 HH1=2.1D-2*WLW/DHY
      HH4=(TTW+2.7315D+2)/(TTS+2.7315D+2)
      HH4=DSQRT(HH4)
CMIS+ HTC=HH1*HH2*HH3*HH4
      HTCD=HH1*HH2*HH3*HH4
CMIS- MODE=9
C
C-----HAUSEN-CORRELATION-----
C-----MODE=9-----
C
      170 HH1=3.66D0*WLW/DHY

```

```

HH2=TTW+2.73152D2
HH3=TTS+2.73152D2
HH2=(HH2/HH3)**2.5D-1
HTCL=HH1*HH2
IF(HTCL.LT.1.D1) HTCL=1.D1
CMIS+ IF(HTCL.GT.HTC) HTC=HTCL
      IF(HTCL.GT.HTCD) HTCD=HTCL
      SQVM=SQV
      IF(SQVM.LT.POL(4)) SQVM=POL(4)
C
C-----
C---BERECHNUNG DER KONDENSATION NACH NUSSELT (MOD. NACH MINCOWYCZ UND
C---ROSENOW-----
C                                     MODE=5
C
      IF(TTS.GT.TSAE) GOTO 175
      TTV=TTW
      IF(TTW.GT.TSAE) TTV=TSAE
      HH1=TTS+3.1D-1*(TTV-TTS)
CIBM+ CALL MPZUS1(DP,DQ,P,HH1,3,1,STOF1,1,STOF2,1,STOF3)
CIBM-
CCVX+
C      CALL MPZUS1_3A(DP,DQ,P,HH1,STOF1,STOF2,STOF3)
CCVX-
      ETA1=STOF1
      CEP1=STOF2
      WLF1=STOF3
C      WRITE(6,*) ETA1,CEP1,WLF1,HH1,TTV,TTS
      HH1=ETA1*HLENGT*WLF1*(TTV-TTS)
      HH2=1.D0/VW
      HH3=1.D0/VD
      HWDX=HWD*(1.D0+6.8D-1*CEP1*(TTV-TTS)/HWD)
      HH4= EB *HWDX*HH2*(HH2-HH3)
      HH5=(HH1/HH4)**2.5D-1
C      WRITE(6,*) HH1,HH4,HH5,HWDX
      DELTA1=1.0606D0*HH5
      DELTA2=5.D-1*DHY*(1.D0-DSQRT(SQVM))
      DELTA=DMAX1(DELTA1,DELTA2)
      DELTA=DELTA/DMAX1(SQVM**1.D-1,1.D-5)
      IF(DELTA.LT.1.D-10) DELTA=1.D-10
      HTCC1=WLF1/DELTA
      RE=DHY*GFW/ETA1
      IF(RE.LT.1.D1) RE=1.D1
      HH1=1.D0/VW
      HH2=1.D0/VD
      HH3=HH1*(HH1-HH2)* EB /ETA1/ETA1
      RE3=1.D0/RE**3.33D-1
      HH3=HH3**3.33D-1
      HTCC4=1.1025D0*RE3*HH3*WLF1
      HTCC=DMIN1(HTCC1,HTCC4)
CMIS HTCC5=(HTCC*(TTV-TTS)+HTC *(TTW-TTS))/(TTW-TTS)
      HTCC5=(HTCC*(TTV-TTS)+HTCD*(TTW-TTS))/(TTW-TTS)
      HH1=TTW
      IF(TTW.LT.TSAE) HH1=TSAE
CIBM+ CALL MPZUS1(DP,DQ,P,HH1,4,1,STOF1,1,STOF2,1,STOF3)
CIBM-
CCVX+
C      CALL MPZUS1_4A(DP,DQ,P,HH1,STOF1,STOF2,STOF3)
CCVX-
      ETA2=STOF1
      CEP2=STOF2
      WLF2=STOF3
      REG=DHY*GFD/ETA2
C+LER
C      IF(REG.LT.1.D0) REG=1.D0
C      REG25=REG**2.5D-1
      REG25=DMAX1(RETURB,REG)**2.5D-1
C-LER
      HH1=1.D0/VD
      HHG=GFD/HH1

```

```

      HH2=DMAX1(VST,HHG)
      HH2=HH2*HH2
      HH3=5.D-1*HH1*HH2
      THA=7.92D-2*HH3/REG25
      PR=CEP1*ETA1/WLF1
      PR5=PR**5.D-1
      THA5=THA**5.D-1
      HH4=1.D0/VW
      HH4=HH4**5.D-1
      HTCC6=6.5D-2*HH4*PR5*THA5*WLF1/ETA1
      HTCC6=HTCC6*(TTV-TTS)/(TTW-TTS)
CMIS+ HTC=DMAX1(HTCC5,HTCC6)
      HTCD=DMAX1(HTCC5,HTCC6)
175 HTC3=HTCD
      HTC=HTCD
C      - FOR MODE=1 OR 5
      IF(SQV.LT.POL(3) .OR. SQH .LE. POL(1)) GO TO 240
      HTC=HTCD
      MODE=9
CMIS-
      IF(SQV.LT.POL(4))GO TO 190
      GO TO 240
190 HH1=SQV
      HH2=POL(3)
      HH3=POL(4)
      HTC1=1.D0/HTC1
      HTC3=1.D0/HTC3
      CALL SINPO (HTC1,HTC3,HH1,HH2,HH3,HTC,MINTER,4)
      HTC=1.D0/HTC
      MODE=6
      GO TO 240
225 HH1=SQH
      HH2=POL(1)
      HH3=POL(2)
      HTC1=1.D0/HTC1
      HTC5=1.D0/HTC5
      CALL SINPO(HTC5,HTC1,HH1,HH2,HH3,HTC,MINTER,1)
      HTC=1.D0/HTC
      MODE=2
CMIS+
C      HTC>0 FOR SQM>0 (NOT DEPENDENT ON POL(1,2) )
      IF (SQM .GT. ZERO .AND. IEQUH .GT. 1) GOTO 150
CMIS-
240 CONTINUE
C
      IF(HTC.LE.ZERO) GOTO 250
C
CSTF 4.12.81 KOPIE VON WAH
      IF (LSSC) HTC=(HTC+HTCK)*0.5D0
CSTF 4.12.81 KOPIE VON WAH
C
      IF (HTCV .GT. HTC) HTCV = HTC
C
      IF(.NOT.LWRITE) RETURN
      WRITE(IOUT,1031)
      WRITE(IOUT,*) NHV,TF,P,SQH,V,VW,VD,HV,HWD,QH,GF,TTW,TSAE,TTS,
      *HTC,HTCK,DHY,HLENGT,(POL(IW),IW=1,4),MODE,
      *(IHTCI(IW),IW=1,3),SQV,GFW,GFD
      WRITE(IOUT,*) '          HTCV = ', HTCV
C
      RETURN
C
250 IF(HTCK.LE.ZERO.OR.HTC.LE.ZERO) WRITE(IOUT,1000) HTCK,HTC
      IF(DHY.LE.ZERO) WRITE(IOUT,1010) DHY
      IF(HLENGT.LE.ZERO) WRITE(IOUT,1020) HLENGT
      WRITE(IOUT,1031)
      WRITE(IOUT,*) NHV,TF,P,SQH,V,VW,VD,HV,HWD,QH,GF,TTW,TSAE,TTS,
      *HTC,HTCK,DHY,HLENGT,(POL(IW),IW=1,4),MODE,
      *(IHTCI(IW),IW=1,3)
      CALL ERRTRA
      STOP
1000 FORMAT('/' W A R N I N G      HTCK = ',1PD13.4,5X,

```

```

      *HTC = ',1PD13.4,5X,'MUST BE .GT. 0.0      STOP IN ----MHTCN1---')
1010 FORMAT(/' W A R N I N G      DHY = ',1PD13.4,5X,'MUST BE .GT. 0.0
      *      STOP IN ----MHTCN1----')
1020 FORMAT(/' W A R N I N G      HLENGT = ',1PD13.4,5X,'MUST BE .GT. 0.
      *0      STOP IN ----MHTCN1----')
1030 FORMAT ('MHTCN1 - NHV,TF,P,SQH,V,VW,VD,HW,HWD,QH,GF,TTW,TSAE,HTC,H
      *TCK,DHY,HLENGT,POL(1...4),MODE,IHTCI(1...3),SQV,GFW,GFD')
1031 FORMAT ('MHTCN1 - NHV,TF,P,SQH,V,VW,VD,HW,HWD,QH,GF,TTW,TSAE,TTS,H
      *TC,HTCK,DHY,HLENGT,POL(1...4),MODE,IHTCI(1...3),SQV,GFW,GFD')
      ENDIF
      END

```

## MHTCN1A

```

      SUBROUTINE MHTCN1A (TF,P,SQH,SQM,SQV,V,VW,VD,HW,HWD,QH,GF,GFD,GFW,
      *      TTS,TTW,TSAE,HTC,HTCK,DHY,HLENGT,POL,MODE,NHV,
      *      IHTCI,LWRITE,VST, HTCVC,IEQUH)
CH+
CN      MHTCN1
CA      POI
CM      LER 21. 2.94: CARPENTER+COLBURN: REYNOLDS-LIMITATION IMPROVED
C*
CV      MOD1.1A
C*
CP      DETERMINATION OF HEAT TRANSFER COEFFICIENTS
CP      FOR HEAT FLOW FROM FLUID TO WALL: TFLUID > TWALL
C*
CI      DHY          : HYDRAULIC DIAMETER OF COOLANT CHANNEL          M
CI      HLENGT       : LENGTH OF HTV
CI      HTCK         : OLD HEAT TRANSFER COEF. (E.G. AT BEGIN OF TIME STEP)
CI                      W/M*M*K
CI      HW           : SPECIFIC ENTHALPY OF SATURATED WATER          J/KG
CI      HWD          : HEAT OF EVAPORATION                          J/KG
CI      IEQUH        : INDICATOR IF SPLITTING OF HEAT FLUX IS DONE
CI                      FOR 2E-CV'S
CI      IHTCI ( )    : INDEX OF CORRELATION TO BE USED FOR DETERMINATION
CI                      OF HTC
CI      LWRITE       : CONTROLS OUTPUT
CI      NHV          : INDEX OF HTV
CI      POL ( )     : INTERPOLATION BOUNDARYS FOR A COSINE FORM
CI                      TRANSITION BETWEEN HTC CORRELATIONS
CI      QH           : HEAT FLUX AT WALL                          W/M*M
CI      SQH          : ENTHALPY VOID FRACTION  $X=H-H'/R$               -
CI      SQV          : VOID FRACTION                              -
CI      TF           : PROBLEM TIME                              S
CI      TSAE         : SATURATION TEMPERATURE                    DEG C
CI      TTS          : SURFACE TEMPERATURE                      DEG C
CI      TTW          : COOLANT TEMPERATURE                      DEG C
CI      V            : SPECIFIC VOLUME OF COOLANT                M*M*M/KG
CI      VD           : SPECIFIC VOLUME OF SATURATED STEAM        M*M*M/KG
CI      VW           : SPECIFIC VOLUME OF SATURATED WATER        M*M*M/KG
CI      SQM          : MASS FLOW QUALITY
C*
CU      GF           : MASS FLUX                                  KG/S*M*M
CU      GFD          : VAPOUR MASS FLUX                          KG/S*M*M
CU      GFW          : WATER MASS FLUX                          KG/S*M*M
CU      HTC          : HEAT TRANSFER COEFF. TO BE CALCULATED      W/M*M*K
CU      MODE         : INDEX OF CORRELATION USED FOR DETERMINATION OF HTC -
CU      P            : COOLANT PRESSURE                          PA
CU      VST          : ABSOLUTE VALUE OF REL. VELOCITY
CU      HTCVC        : HEAT TRANSFER COEF. FOR WALL - VAPOR HEAT TRANSFER
CU                      W/M*M*K
C*
CF      MPTAF
C*
CR      ERRTRA      MPSPLK      MPZUS1      SINPO
C*
CB      MHTCN      MHTCQF

```

```

C*
CH-
    INCLUDE (C)
    INCLUDE (CA)
    INCLUDE (CDSS16)
    INCLUDE (CSEBPI)
CAS---Common-Blocks fuer KONWAR
C    INCLUDE (CHI0)
    INCLUDE (CDGE28)
    INCLUDE (CDNW55)
    INCLUDE (CHNW02)
    INCLUDE (ANDREA)
CAS---ENDE Common-Blocks fuer KONWAR
C
    DIMENSION POL (000014),IHTCI (000003)
CIBM+
    DOUBLE PRECISION MPTAF, TFA
CAS---Variablendefinition fuer KONWAR
    DOUBLE PRECISION S0I, WERT
    INTEGER          MODEW, IHV, IZAEHL
CIBM-
CCVX+
C    DOUBLE PRECISION MPTAF_P101,MPTAF_P104
CCVX-
    LOGICAL LWRITE, LWRT1
C
C    PARAMETER (KPI = 3.14159)
    PARAMETER (ZERO = 0.D0)
    PARAMETER (ONE = 1.D0)
    PARAMETER (IOUT = 6)
    PARAMETER (RETURB = 5.D3)
C
    DATA TFA /0./
    DATA IZAEHL /0/
C    WRITE(66,*) 'QH nach Uebergabe ', QH
C
CMIS+
C    DEFAULT VALUE FOR HTC
    HTC=ZERO
CMIS-
C
C    OUTPUTAKTIVIERUNG alle 25 Sekunden
    IF (TF-TFA .GE.25.0) THEN
        WRITE(66,*) 'MHTCNlas.F Problemzeit = ', TF
        LWRT1=.TRUE.
        TFA=TF
        IZAEHL=1
    ELSE
        IZAEHL=IZAEHL+1
        IF (IZAEHL.LE.4) THEN
            LWRT1=.TRUE.
        ELSE
            LWRT1=.FALSE.
        ENDIF
    ENDIF
    LWRT1=.FALSE.
    IF (LWRT1) THEN
        WRITE (66,*) ' MHTCNlas - Aufruf'
        WRITE (66,*) ' NHV = ', NHV, ' SQH = ', SQH, ' SQV = ', SQV
        WRITE (66,*) ' POL(1) = ', POL(1), ' POL(2) = ', POL(2),
        *          ' POL(3) = ', POL(3), ' POL(4) = ', POL(4)
    ENDIF
C
CMIS+
C    DEFAULT VALUE FOR HTC
    HTC=ZERO
    MODEW = 0
CMIS-
C
C    CHECK INPUT DATA
    IF (HTCK.LE.ZERO.OR.DHY.LE.ZERO.OR.HLENGT.LE.ZERO) GOTO 250
    IF (.NOT.LWRITE) GOTO 10
C
    WRITE(66,1030)

```

```

C      WRITE(IOUT,*) NHV,TF,P,SQH,V,VW,VD,HW,HWD,QH,GF,TTW,TSAE,
C      *HTC,HTCK,DHY,HLENGT,(POL(IW),IW=1,4),MODE,
C      *(IHTCI(IW),IW=1,3),SQV,GFW,GFD
C
CSTF 4.12.81
  10 CONTINUE
    IF(GF.LT.1.D-10) GF=1.D-10
    IF(GFW.LT.1.D-10) GFW=1.D-10
    IF(GFD.LT.1.D-10) GFD=1.D-10
C
    HTCMAC=ZERO
    HTCC=ZERO
C
    CALL MPSPLK(P,DP,DQ)
C
C
    IF (SQV.GE.POL(4)) GOTO 150
C
C-----
C*****SUBCOOLED FORCED CONVECTION*****
C-----
C
C-----DITTUS-BOELTER CORRELATION I-----
C-----MODE=1-----
C
    PH1=P
    HH1=(TTS+TTW)*5.D-1
CIBM+
    CALL MPZUS1(DP,DQ,PH1,HH1,3,1,ETAW,1,CEPW,1,WLFW)
CIBM-
CCVX+
C    CALL MPZUS1_3A(DP,DQ,PH1,HH1,ETAW,CEPW,WLFW)
CCVX-
    ETA1=ETAW
    CEP1=CEPW
    WLF1=WLFW
    IF (SQH.GE.POL(2)) GOTO 60
    SQHM=SQH
    IF(SQH.GT.POL(1)) SQHM=POL(1)
    RE=DHY*GFW/ETA1
C    WRITE(6,*) ETA1,CEP1,WLF1,ETA2,RE
    H19=SQHM
    IF(SQHM.LT.1.D-6) SQHM=1.D-6
    IF(SQHM.GT.9.9999D-1) SQHM=9.9999D-1
    REW=(ONE-SQHM)*RE
    RE8=REW*0.8D0
    PR=CEP1*ETA1/WLF1
    PR4=PR*0.4D0
    HTC5=0.023D0*RE8*PR4*(WLF1/DHY)
C    WRITE(6,*) HTC5,RE8,PR4,WLF1,DHY,FCH
    SQHM=H19
    MODE=1
C
  60 CONTINUE
C
C-----NATURAL CONVECTION -----
    TFILM=(TTW+TTS)*5.D-1
    IF(TFILM.GT.TSAE) TFILM=TSAE
CIBM+
    CALL MPZUS1(DP,DQ,PH1,TFILM,3,1,ETAW,1,CEPW,1,WLFW)
CIBM-
CCVX+
C    CALL MPZUS1_3A(DP,DQ,PH1,TFILM,ETAW,CEPW,WLFW)
CCVX-
    PR=CEPW*ETAW/WLFW
CIBM+
    RHOWF=1.D0/MPTAF(101,PH1,TFILM)
    BETA=MPTAF(104,PH1,TFILM)
CIBM-
CCVX+
C    RHOWF=1.D0/MPTAF_P101(PH1,TFILM)
C    BETA=MPTAF_P104(PH1,TFILM)

```

```

CCVX-   GRAS=RHOWF*RHOWF* EB *DABS((TTS-TTW)*BETA)/(ETAW*ETAW)
C       WRITE(6,*) NHV,RHOWF,ETAW,BETA,TTS,TTW
        HH1=GRAS*PR
        HH1=HH1**0.3333D0
        EXP1=-9.D0/1.6D1
        EXP2=-16.D0/27.D0
        FPR=(1.D0+(2.D0*PR)**EXP1)**EXP2
        HTCL=0.15D0*WLFW*HH1*FPR
        IF(HTCL.LT.2.0D1) HTCL=2.0D1
        IF(SQH.GE.POL(2)) GOTO 61
        IF(HTCL.GT.HTC5) HTC5=HTCL
        IF(SQH.GT.POL(1)) GOTO 61
        HTC=HTC5
CMIS+
C       HTC>0 FOR SQM>0 (NOT DEPENDENT ON POL(1,2) )
        IF (SQM .GT. ZERO .AND. IEQUH .GT. 1) GOTO 150
CMIS-
        GO TO 240
C-----
C----SUBCOOLED AND SATURATED NUCLEATE BOILING-----
C-----
C----BERECHNUNG VON HCMAC NACH CHEN-----
C-----
C-----MODE=5-----
C
C LER IF CONDENSATION ON WALL: NO MICROSCOPIC HEAT TRANSFER
61 SQHM=SQH
   IF(SQH.LT.POL(2)) SQHM=POL(2)
   SQHM3=POL(3)/VD/((1.D0-POL(3))/VW+POL(3)/VD)
   IF(SQH.GT.SQHM3) SQHM=SQHM3
   PH1=P
   HH1=(TTS+TTW)*5.D-1
   IF(SQH.GE.ZERO) GO TO 70
CIBM+
        CALL MPZUS1(DP,DQ,PH1,HH1,3,1,ETAW,1,CEPW,1,WLFW)
CIBM-
CCVX+
C       CALL MPZUS1_3A(DP,DQ,PH1,HH1,ETAW,CEPW,WLFW)
CCVX-
        ETA1=ETAW
        CEP1=CEPW
        WLF1=WLFW
        GO TO 80
CIBM+
70 CALL MPZUS1(DP,DQ,PH1,HH1,1,1,STOF1,1,STOF2,1,STOF3)
CIBM-
CCVX+
C 70 CALL MPZUS1_1A(DP,DQ,PH1,HH1,STOF1,STOF2,STOF3)
CCVX-
        ETA1=STOF1
        CEP1=STOF2
        WLF1=STOF3
C
CIBM+
80 CALL MPZUS1(DP,DQ,P,HH1,2,1,STOF1,1,STOF2,1,STOF3)
CIBM-
CCVX+
C 80 CALL MPZUS1_2A(DP,DQ,P,HH1,STOF1,STOF2,STOF3)
CCVX-
        ETA2=STOF1
        CEP2=STOF2
        WLF2=STOF3
        RE=DHY*GF/ETA1
C       WRITE(6,*) ETA1,CEP1,WLF1,RE
C !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
        IF(RE.LT.1.0D+2) RE=1.0D+2
C !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
        H19=SQHM
        IF(SQHM.LT.1.D-6) SQHM=1.D-6
        IF(SQHM.GT.9.9999D-1) SQHM=9.9999D-1
        REW=(ONE-SQHM)*RE
        RE8=REW**0.8D0

```



```

PR=CEP1*ETA1/WLF1
PR4=PR**0.4D0
HH1=(ETA2/ETA1)**0.1D0
HH2=(VD/VW)**0.5D0
HH3=(SQHM/(ONE-SQHM))**0.9D0
HH1=HH1*HH2*HH3
HH1=HH1**0.8174D0
FCH=ONE+1.6D0*(HH1)
HTCMAC=0.023D0*RE8*PR4*(WLF1/DHY)*FCH
IF(HTCMAC.LT.HTCL) HTCMAC=HTCL
SQHM=H19
C-----
C----BERECHNUNG DER KONDENSATION NACH NUSSELT (MOD. NACH MINCOWYCZ UND -
C----ROSENOW-----
C-----MODE=5-----
C
SQVM=SQV
IF (LWRIT1) THEN
WRITE (66,*) ' KONWAR EINBAU OBEN, VOR KONWAR AUFRUF'
WRITE (66,*) ' POL(1) = ', POL(1), ' POL(2) = ', POL(2),
* ' POL(3) = ', POL(3), ' POL(4) = ', POL(4)
WRITE(66,*) ' SQV = ', SQV, ' SQVM = ', SQVM,
* ' SQH = ', SQH, ' SQHM = ', SQHM
WRITE(66,*) ' IEQUH = ', IEQUH, ' TTS = ', TTS,
* ' TSAE = ', TSAE, ' TTW = ', TTW
ENDIF
IF(SQVM.GT.POL(3)) SQVM=POL(3)
IF(TTS.GT.TSAE) GOTO 85
TTV=TTW
IF(TTW.GT.TSAE) TTV=TSAE
HH1=TTS+3.1D-1*(TTV-TTS)
C ErweiterungKONWAR-----27.1.1995
IF (LWRIT1) THEN
WRITE (66,*) 'EINBAUOBEN'
WRITE (66,*) 'NHV = ', NHV, ' NEIGUNG = ', ANEIG(NHV)
ENDIF
CALL KONWAR (SQM,SQV,GF,P,TTW,TTV,TTS,DHY,HLENGT,MODEW,
* WERT, LWRIT1)
HTC = WERT
IF (LWRIT1) THEN
WRITE (66,*) ' AUFRUF KONWAR OBEN'
WRITE (66,*) ' MODEW = ', MODEW
WRITE (66,*) ' HTC = ', HTC
WRITE (66,*) ' @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@'
ENDIF
GOTO 85
C KONWAR-Erweiterung-----
CIBM+
CALL MPZUS1(DP,DQ,P,HH1,3,1,STOF1,1,STOF2,1,STOF3)
CIBM-
CCVX+
C CALL MPZUS1_3A(DP,DQ,P,HH1,STOF1,STOF2,STOF3)
CCVX-
ETA1=STOF1
CEP1=STOF2
WLF1=STOF3
C WRITE(6,*) ETA1,WLF1,TTW
HH1=ETA1*HLENGT*WLF1*(TTV-TTS)
HH2=1.D0/VW
HH3=1.D0/VD
HWDX=HWD*(1.D0+6.8D-1*CEP1*(TTV-TTS)/HWD)
HH4= EB *HWDX*HH2*(HH2-HH3)
HH5=(HH1/HH4)**2.5D-1
C WRITE(6,*) HH1,HH4,HH5
DELTA1=1.0606D0*HH5
DELTA2=5.D-1*DHY*(1.D0-DSQRT(SQVM))
DELTA=DMAX1(DELTA1,DELTA2)
DELTA=DELTA/DMAX1(SQVM**1.D-1,1.D-5)
IF(DELTA.LT.1.D-10) DELTA=1.D-10
HTCC1=WLF1/DELTA
C ----- NUSSELT -----
RE=DHY*GFW/ETA1

```

```

IF(RE.LT.1.D1) RE=1.D1
IF(RE.GT.2.3D+3) RE=2.3D+3
HH1=1.D0/VW
HH2=1.D0/VD
HH3=HH1*(HH1-HH2)* EB /ETA1/ETA1
RE3=1.D0/RE**3.33D-1
HH3=HH3**3.33D-1
HTCC3=1.1025D0*RE3*HH3*WLF1
HTCC4=DMIN1(HTCC1,HTCC3)
HTCC5=HTCC4*(TTV-TTS)/(TTW-TTS)
HTCC5=DMAX1(HTCMAC,HTCC5)
C ----- CARPENTER UND COLBURN -----
HH1=TTW
IF(TTW.LT.TSAE) HH1=TSAE
CIBM+
CALL MPZUS1(DP,DQ,P,HH1,4,1,STOF1,1,STOF2,1,STOF3)
CIBM-
CCVX+
C CALL MPZUS1_4A(DP,DQ,P,HH1,STOF1,STOF2,STOF3)
CCVX-
ETA2=STOF1
CEP2=STOF2
WLF2=STOF3
REG =DHY*GFD/ETA2
C+LER
REG25=DMAX1(RETURB,REG)**2.5D-1
C-LER IF(REG25.LT.1.D0) REG25=1.D0
HH1=1.D0/VD
HHG=GFD/HH1
HH2=DMAX1(VST,HHG)
HH2=HH2*HH2
HH3=5.D-1*HH1*HH2
THA=7.92D-2*HH3/REG25
PR=CEP1*ETA1/WLF1
PR5=PR**5.D-1
THA5=THA**5.D-1
HH4=1.D0/VW
HH4=HH4**5.D-1
HTCC6=6.5D-2*HH4*PR5*THA5*WLF1/ETA1
HTCC6=HTCC6*(TTV-TTS)/(TTW-TTS)
HTC=DMAX1(HTCC5,HTCC6)
C
85 HTC1=HTC
MODE=5
CAS--Modekorrektur fuer KONWAR
MODE = 100 * MODE + MODEW
IF (LWRIT1) THEN
WRITE (66, *) 'MODE nach Korrektur in KONWAR = ', MODE
ENDIF
CAS---ENDE Modekorrektur
IF(SQH.LT.POL(2)) GOTO 225
IF (SQV.GT.POL(3))GO TO 150
CMIS+
IF (IEQUH .GT. 1) GO TO 150
CMIS-
GO TO 240
C
C-----
C*****SINGLE PHASE STEAM*****
C-----
C
C-----DITTUS-BOELTER-CORRELATION-II-----
C-----MODE=9-----
C
150 PH1=P
HH1=(TTW+TTS)*5.D-1
C
CSTF 4.12.81 KOPIE VON WAH
IF (HH1.LT.TSAE) HH1=TSAE
CSTF 4.12.81 KOPIE VON WAH
C
CIBM+

```

```

      CALL MPZUS1(DP,DQ,PH1,HH1,4,1,ETAD,1,CEPD,1,WLFD)
CIBM-
CCVX+
C      CALL MPZUS1_4A(DP,DQ,PH1,HH1,ETAD,CEPD,WLFD)
CCVX-
      ETA=ETAD
      CEP=CEPD
      WLW=WLFD
      RE=DHY*GFD/ETA
      IF(RE.LT.5.D3) RE=5.D3
C      WRITE(6,*) ETA,CEP,WLW,RE
      HH2=RE**8.D-1
      PR=CEP*ETA/WLW
      HH3=PR**4.D-1
C
      IF(IHTCI(3).EQ.2) GOTO 160
      HH1=2.3D-2*WLW/DHY
CMIS+ HTC =HH1*HH2*HH3
      HTCD=HH1*HH2*HH3
CMIS- MODE=9
C      WRITE(6,*) HTC,HH1,HH2,HH3,PR
      GOTO 170
C
C-----MC ELIGOT-----
C-----IHTCI(3)=2-----MODE=9-----
C
      160 HH1=2.1D-2*WLW/DHY
      HH4=(TTW+2.7315D+2)/(TTS+2.7315D+2)
      HH4=DSQRT(HH4)
CMIS+ HTC=HH1*HH2*HH3*HH4
      HTCD=HH1*HH2*HH3*HH4
CMIS- MODE=9
C
C-----HAUSEN-CORRELATION-----
C-----MODE=9-----
C
      170 HH1=3.66D0*WLW/DHY
      HH2=TTW+2.7315D2
      HH3=TTS+2.7315D2
      HH2=(HH2/HH3)**2.5D-1
      HTCL=HH1*HH2
      IF(HTCL.LT.1.D1) HTCL=1.D1
CMIS+ IF(HTCL.GT.HTC) HTC=HTCL
      IF(HTCL.GT.HTCD) HTCD=HTCL
      SQVM=SQV
      IF(SQVM.LT.POL(4)) SQVM=POL(4)
C
C-----
C-----BERECHNUNG DER KONDENSATION NACH NUSSELT (MOD. NACH MINCOWYCZ UND -
C-----ROSENOW-----
C-----MODE=5-----
C
      IF(TTS.GT.TSAE) GOTO 175
      TTV=TTW
      IF(TTW.GT.TSAE) TTV=TSAE
      HH1=TTS+3.1D-1*(TTV-TTS)
CAS--Einbau KONWAR-----27.1.1995
      MODEW = 0
      IF (LWRIT1) THEN
        WRITE (66,*) ' KONWAR EINBAU UNTEN, VOR KONWAR AUFRUF '
        WRITE (66,*) ' POL(1) = ', POL(1), ' POL(2) = ', POL(2),
        * ' POL(3) = ', POL(3), ' POL(4) = ', POL(4)
        WRITE(66,*) ' SQV = ', SQV, ' SQVM = ', SQVM,
        * ' SQH = ', SQH, ' SQHM = ', SQHM
        WRITE(66,*) ' IEQUH = ', IEQUH, ' TTS = ', TTS,
        * ' TSAE = ', TSAE, ' TTW = ', TTW
      ENDIF
      IF (LWRIT1) THEN
        WRITE (66,*) ' EINBAUUNTEN '
        WRITE (66,*) ' NHV = ', NHV, ' NEIGUNG = ', ANEIG(NHV)
      ENDIF
      CALL KONWAR (SQM,SQV,GF,P,TTW,TTV,TTS,DHY,HLENGT,MODEW,

```

```

*          WERT, LWRT1)
  HTCD = WERT
  IF (LWRT1) THEN
    WRITE (66, *) ' AUFRUF KONWAR UNTEN'
    WRITE (66, *) ' MODEW = ', MODEW, 'MODE = ', MODE
    WRITE (66, *) ' HTCD = ', HTCD
    WRITE (66, *) ' @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@'
  ENDIF
  GOTO 175
CAS---ENDE EINBAU KONWAR-----
CIBM+
  CALL MPZUS1(DP,DQ,P,HH1,3,1,STOF1,1,STOF2,1,STOF3)
CIBM-
CCVX+
C    CALL MPZUS1_3A(DP,DQ,P,HH1,STOF1,STOF2,STOF3)
CCVX-
  ETA1=STOF1
  CEP1=STOF2
  WLF1=STOF3
C    WRITE(6,*) ETA1,CEP1,WLF1,HH1,TTV,TTS
  HH1=ETA1*HLENGT*WLF1*(TTV-TTS)
  HH2=1.D0/VW
  HH3=1.D0/VD
  HWDX=HWD*(1.D0+6.8D-1*CEP1*(TTV-TTS)/HWD)
  HH4= EB *HWDX*HH2*(HH2-HH3)
  HH5=(HH1/HH4)**2.5D-1
C    WRITE(6,*) HH1,HH4,HH5,HWDX
  DELTA1=1.0606D0*HH5
  DELTA2=5.D-1*DHY*(1.D0-DSQRT(SQVM))
  DELTA=DMAX1(DELTA1,DELTA2)
  DELTA=DELTA/DMAX1(SQVM**1.D-1,1.D-5)
  IF(DELTA.LT.1.D-10) DELTA=1.D-10
  HTCC1=WLF1/DELTA
  RE=DHY*GFW/ETA1
  IF(RE.LT.1.D1) RE=1.D1
  HH1=1.D0/VW
  HH2=1.D0/VD
  HH3=HH1*(HH1-HH2)* EB /ETA1/ETA1
  RE3=1.D0/RE**3.33D-1
  HH3=HH3**3.33D-1
  HTCC4=1.1025D0*RE3*HH3*WLF1
  HTCC=DMIN1(HTCC1,HTCC4)
CMIS  HTCC5=(HTCC*(TTV-TTS)+HTC *(TTW-TTS))/(TTW-TTS)
      HTCC5=(HTCC*(TTV-TTS)+HTCD*(TTW-TTS))/(TTW-TTS)
      HH1=TTW
      IF(TTW.LT.TSAE) HH1=TSAE
CIBM+
  CALL MPZUS1(DP,DQ,P,HH1,4,1,STOF1,1,STOF2,1,STOF3)
CIBM-
CCVX+
C    CALL MPZUS1_4A(DP,DQ,P,HH1,STOF1,STOF2,STOF3)
CCVX-
  ETA2=STOF1
  CEP2=STOF2
  WLF2=STOF3
  REG=DHY*GFD/ETA2
C+LER
C    IF(REG.LT.1.D0) REG=1.D0
C    REG25=REG**2.5D-1
  REG25=DMAX1(RETURB,REG)**2.5D-1
C-LER
  HH1=1.D0/VD
  HHG=GFD/HH1
  HH2=DMAX1(VST,HHG)
  HH2=HH2*HH2
  HH3=5.D-1*HH1*HH2
  THA=7.92D-2*HH3/REG25
  PR=CEP1*ETA1/WLF1
  PR5=PR**5.D-1
  THA5=THA**5.D-1
  HH4=1.D0/VW
  HH4=HH4**5.D-1

```

```

      HTCC6=6.5D-2*HH4*PR5*THA5*WLF1/ETA1
      HTCC6=HTCC6*(TTV-TTS)/(TTW-TTS)
CMIS+ HTC=DMAX1(HTCC5,HTCC6)
      HTCD=DMAX1(HTCC5,HTCC6)
175 HTC3=HTCD
      HTC=HTCD
C      - FOR MODE=1 OR 5
      IF(SQV.LT.POL(3) .OR. SQH .LE. POL(1)) GO TO 240
      HTC=HTCD
      MODE=9
CAS---Modekorrektur fuer KONWAR
      MODE = 100 * (MODE+10) + MODEW
      IF (LWRIT1) THEN
        WRITE (66, *) 'MODE nach Korrektur in KONWAR = ', MODE
      ENDIF
CMIS-
      IF(SQV.LT.POL(4))GO TO 190
      GO TO 240
190 HH1=SQV
      HH2=POL(3)
      HH3=POL(4)
      HTC1=1.D0/HTC1
      HTC3=1.D0/HTC3
      CALL SINPO (HTC1,HTC3,HH1,HH2,HH3,HTC,MINTER,4)
      HTC=1.D0/HTC
CAS ALTE ATHLETVERSION      MODE=6
      MODE = 1600 + MODEW
CAS---Modekorrektur fuer KONWAR
      MODE = 100 * (MODE+10) + MODEW
      IF (LWRIT1) THEN
        WRITE (66, *) 'MODE nach Korrektur in KONWAR = ', MODE
      ENDIF
      GO TO 240
225 HH1=SQH
      HH2=POL(1)
      HH3=POL(2)
      HTC1=1.D0/HTC1
      HTC5=1.D0/HTC5
      CALL SINPO(HTC5,HTC1,HH1,HH2,HH3,HTC,MINTER,1)
      HTC=1.D0/HTC
CAS ALTE ATHLETVERSION      MODE=2
      MODE = 1200 + MODEW
CAS---Modekorrektur fuer KONWAR
      IF (LWRIT1) THEN
        MODE = 100 * (MODE +10) + MODEW
        WRITE (66, *) 'MODE nach Korrektur in KONWAR = ', MODE
      ENDIF
CMIS+
C      HTC>0 FOR SQM>0 (NOT DEPENDENT ON POL(1,2) )
      IF (SQM .GT. ZERO .AND. IEQUH .GT. 1) GOTO 150
CMIS-
240 CONTINUE
C
      IF(HTC.LE.ZERO) GOTO 250
C
CSTF 4.12.81 KOPIE VON WAH
      IF (LSSC) HTC=(HTC+HTCK)*0.5D0
CSTF 4.12.81 KOPIE VON WAH
C
      IF (HTCV .GT. HTC) HTC=HTCV
C
      IF(.NOT.LWRITE) RETURN
      WRITE(66,1031)
      WRITE(IOUT,*) NHV,TF,P,SQH,V,VW,VD,HV,HWD,QH,GF,TTW,TSAE,TTS,
C      *HTC,HTCK,DHY,HLENGT,(POL(IW),IW=1,4),MODE,
C      *(IHTCI(IW),IW=1,3),SQV,GFWD,GF
C      WRITE(IOUT,*) '          HTC = ', HTC
C
      RETURN
C
250 IF(HTCK.LE.ZERO.OR.HTC.LE.ZERO) WRITE(IOUT,1000) HTCK,HTC
      IF(DHY.LE.ZERO) WRITE(IOUT,1010) DHY

```

```

        IF(HLENGT.LE.ZERO) WRITE(IOUT,1020) HLENGT
        WRITE(66,1031)
C      WRITE(IOUT,*) NHV,TF,P,SQH,V,VW,VD,HW,HWD,QH,GF,TTW,TSAE,TTS,
C      *HTC,HTCK,DHY,HLENGT,(POL(IW),IW=1,4),MODE,
C      *(IHTCI(IW),IW=1,3)
        CALL ERRTRA
        STOP
1000 FORMAT(/'  W A R N I N G      HTCK = ',1PD13.4,5X,
*'HTC = ',1PD13.4,5X,'MUST BE .GT. 0.0      STOP IN ----MHTCN1----')
1010 FORMAT(/'  W A R N I N G      DHY = ',1PD13.4,5X,'MUST BE .GT. 0.0
*      STOP IN ----MHTCN1----')
1020 FORMAT(/'  W A R N I N G      HLENGT = ',1PD13.4,5X,'MUST BE .GT. 0.
*0      STOP IN ----MHTCN1----')
1030 FORMAT ('MHTCN1 - NHV,TF,P,SQH,V,VW,VD,HW,HWD,QH,GF,TTW,TSAE,HTC,H
*TCK,DHY,HLENGT,POL(1...4),MODE,IHTCI(1...3),SQV,GFW,GFD')
1031 FORMAT ('MHTCN1 - NHV,TF,P,SQH,V,VW,VD,HW,HWD,QH,GF,TTW,TSAE,TTS,H
*TC,HTCK,DHY,HLENGT,POL(1...4),MODE,IHTCI(1...3),SQV,GFW,GFD')
        END

```



Forschungszentrum Jülich



Jül-3343  
Januar 1997  
ISSN 0944-2952